



## Rechercher parmi ses pairs ou quand le hasard ne fait pas si bien les choses

Etienne Rivière, Philippe Gauron

### ► To cite this version:

Etienne Rivière, Philippe Gauron. Rechercher parmi ses pairs ou quand le hasard ne fait pas si bien les choses. MajecSTIC 2005: Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC, IRISA – IETR – LTSI, Nov 2005, Rennes, pp.37-56. inria-00000672

**HAL Id: inria-00000672**

**<https://inria.hal.science/inria-00000672>**

Submitted on 14 Nov 2005

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Rechercher parmi ses pairs ou quand le hasard ne fait pas si bien les choses

Etienne Rivière<sup>†</sup> et Philippe Gauron<sup>‡</sup>

<sup>†</sup> IRISA/IFSIC (Université de Rennes I), France, [etienne.riviere@irisa.fr](mailto:etienne.riviere@irisa.fr)

<sup>‡</sup> LRI/Université Paris-Sud (Paris XI), France, [philippe.gauron@lri.fr](mailto:philippe.gauron@lri.fr)

**Résumé :** Ce tutoriel est axé sur la recherche de données dans des systèmes répartis déployés à large échelle. Nous y présentons les réseaux pair-à-pair, ainsi que les mécanismes de recherche qui y sont associés. Ces réseaux présentent des caractéristiques communes avec les réseaux d'interaction, objet d'études dans de nombreuses disciplines. Nous montrons que ces propriétés sont liées à l'application et comment en tirer parti pour concevoir des systèmes pair-à-pair.

**Mots-clés :** Systèmes distribués, Réseaux d'interaction, Réseaux pair-à-pair, Réseaux logiques.

## 1 INTRODUCTION

Ce tutoriel s'intéresse aux algorithmes de recherche dans des réseaux organisés selon le modèle pair-à-pair. Si les réseaux pair-à-pair ont de multiples domaines d'applications<sup>1</sup>, nous fondons ce tutoriel sur les applications de partage de données réparties, et plus particulièrement sur une des problématiques centrales de ces réseaux : la recherche et la localisation. Notre étude s'intéresse aussi aux graphes d'interaction, qui modélisent ici les relations entre les utilisateurs du système de partage de données. Ils présentent des propriétés communes avec d'autres modélisations issues de domaines variés ; nous montrons comment ces propriétés peuvent être prises en compte pour améliorer l'efficacité d'un service de partage de données. Nous introduisons enfin l'observation de tels réseaux et l'utilisation des données récoltées dans le cadre des mécanismes de recherche présentés.

Ce tutoriel est organisé comme suit : la partie 2 décrit le modèle utilisé, les motivations qui président au modèle pair-à-pair ainsi que la problématique générale de la recherche et de la localisation dans ces réseaux. La partie 3 présente deux modes d'organisation de tels réseaux, les parties 4 et 5 expliquant le fonctionnement des mécanismes de recherche associée. Les réseaux d'interaction et leurs propriétés sont introduits dans la partie 6, la partie 7 explique ensuite la prise en compte de ces propriétés lors de la définition de mécanismes de recherche de données dans les réseaux pair-à-pair. Enfin, la partie 8 présente l'observation des réseaux pair-à-pair : son objectif, sa mise en œuvre et l'utilisation des données récoltées. Nous présentons en conclusion quelques perspectives d'études.

<sup>1</sup>Citons le calcul réparti, la mutualisation de services ou encore les services de téléphonie.

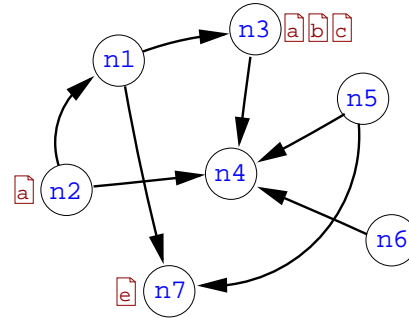


FIG. 1 – Exemple d'un réseau simple

## 2 MODÈLE

Pour représenter et étudier les systèmes répartis, il est nécessaire de les modéliser. De tels systèmes sont représentés par un ensemble de nœuds : ce sont les acteurs du réseau, qu'ils soient des instances d'un programme sur une machine ou des personnes dans le cadre d'un réseau social. Dans ce document, l'ensemble des nœuds d'un réseau sera noté  $\mathcal{N}$ . La communication dans le réseau se fait par envoi de messages entre nœuds. L'arête  $n_1 \rightarrow n_2$  signifie que  $n_1$  connaît l'adresse de  $n_2$ . Un message peut être envoyé d'un nœud  $n_1$  à un nœud  $n_2$  si et seulement si l'arête  $n_1 \rightarrow n_2$  existe. Cette arête est dirigée, c'est à dire qu'une arête  $n_1 \rightarrow n_2$  ne suppose pas une arête  $n_2 \rightarrow n_1$ . Une arête dirigée est appelée une *arc*. La figure 1 montre un exemple de réseau simple, avec 7 nœuds et 8 arcs.

**Réseau physique et réseau logique** Il est important de faire la distinction entre le réseau physique (le support d'envoi des messages comme le sont l'infrastructure d'Internet ou les services postaux) et le réseau logique.

On suppose que ce réseau physique est connexe. Dans un réseau connexe, il est toujours possible de trouver un chemin (une suite d'arêtes) entre toute paire de nœuds du réseau. Ceci impose que si un nœud  $n_a$  connaît l'adresse d'un nœud  $n_b$  il lui est toujours possible d'envoyer un message de  $n_a$  à  $n_b$ <sup>2</sup>. Par exemple, dans le réseau IP (Internet) il est toujours possible, lorsqu'on connaît l'adresse IP et le port d'un nœud, de lui envoyer un message.

Le réseau logique est comme son nom anglais l'indique (*overlay*) superposé au réseau physique. Il est

<sup>2</sup>...éventuellement en suivant un chemin, c'est-à-dire une suite d'arêtes.

constitué des liens entre nœuds fondés sur la connaissance des adresses dans le réseau physique. Un exemple simple de réseau logique est le réseau des connaissances téléphoniques : bien qu'il soit possible à tout participant d'appeler tout autre, les appels se font uniquement vers les nœuds (identifiés par leur numéro) présents dans le répertoire téléphonique de l'appelant.

C'est le réseau logique qui sera considéré pour la transmission de demandes de données et qui transmettra la demande. Une fois trouvé un destinataire, la réponse peut être transmise directement au demandeurs qui aura indiqué son adresse physique dans sa demande (propagée, elle, sur le réseau logique).

**Données et méta-données** L'objectif des systèmes répartis considérés est le partage de ressources mutualisées. Nous nous intéressons à l'utilisation des mécanismes de recherche et de localisation de ces ressources dans ces systèmes. Nous considérons ces ressources comme des objets (de type quelconque) représentés par un identifiant (une lettre minuscule sur les schémas). Chaque ressource est proposée par au moins un nœud du réseau (potentiellement plusieurs, selon les réseaux logiques). Dans la figure 1 la donnée  $a$  est possédée par les nœuds  $n_2$  et  $n_3$ . Chaque donnée est associée à un ensemble plus ou moins riche de méta-données. Ces informations permettent de distinguer l'objet parmi l'ensemble des données. Elles peuvent être de deux types : (i) identification unique : à la ressource est associé un numéro unique dans un espace de nommage ; (ii) des informations non uniques qui peuvent décrire la sémantique (ce qui est représenté par l'objet) ou le format de la donnée : taille de fichier, type de *media* ...

**Quelques propriétés** Afin d'étudier les propriétés de ces réseaux, il est nécessaire d'introduire quelques notions simples utilisées en analyse des graphes. À un moment de son évolution, notre réseau est modélisé par un graphe (défini comme le couple d'ensembles  $(\mathcal{N}, \mathcal{A})$ , ensembles des nœuds et des arêtes).

- **Degré d'un nœud** Le degré d'un nœud  $n$  est, dans un graphe orienté, considéré comme la somme de deux composantes : degré entrant et degré sortant. Le degré entrant est le nombre d'arêtes qui arrivent vers  $n$  et le degré sortant le nombre d'arêtes qui partent de  $n$ . Dans l'exemple de la figure 1, le nœud  $n_1$  a un degré entrant de 1 et un degré sortant de 2.
- **Plus court chemin** La longueur d'un chemin entre deux nœuds est le nombre de liens qu'il faut emprunter pour se rendre d'un nœud à l'autre. La plus court chemin est le chemin qui relie un couple de nœuds et qui emprunte un nombre d'arêtes minimal.
- **Diamètre** Le diamètre d'un graphe est le plus long des chemins parmi l'ensemble des plus courts chemins pour tout couple de nœuds dans le graphe.
- **Distance moyenne** La distance moyenne est la moyenne des longueurs des plus courts chemins entre tous les couples distincts de nœuds d'un graphe.

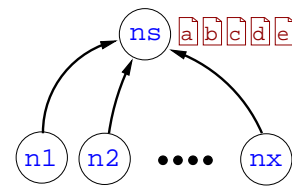


FIG. 2 – Système centralisé

- **Coefficient d'agrégation** Le coefficient d'agrégation sert à répondre à la question suivante pour les nœuds du graphe considéré :

*mes voisins sont-ils voisins entre eux ?*

Le coefficient d'agrégation  $\psi_n$ , pour un nœud  $n$ , est la probabilité pour deux voisins  $n_1$  et  $n_2$  de  $n$  que  $n_1$  et  $n_2$  soient eux-même voisins entre eux. Le coefficient d'agrégation  $\psi$  pour un graphe est donc la moyenne des coefficients d'agrégation pour l'ensemble des nœuds du graphe. Si  $n_{voisins}$  est l'ensemble des voisins de  $n$ , pour tous les couples de voisins possibles, on vérifie si ils sont eux-même voisins. La définition suivante est pour un graphe orienté :

$$\psi_n = \frac{\sum_{a \neq b, (a,b) \in n_{voisins}} \begin{cases} 1 \text{ si } b \in a_{voisins} \\ 0 \text{ sinon} \end{cases}}{|n_{voisins}| \times (|n_{voisins}| - 1)}.$$

Le coefficient d'agrégation permet de déterminer si le voisinage local des nœuds du graphe est localement dense, c'est à dire s'il existe des communautés locales plus connectées que ne l'est le reste du graphe (présence de « communautés »).

## 2.1 Systèmes centralisés

Une solution simple pour proposer l'accès à des objets dans un réseau est de donner la responsabilité de tous les objets au même nœud, qui est le serveur pour tous les autres. L'ensemble des nœuds du réseau connaissent l'adresse de ce nœud centralisateur, et agissent donc comme clients de celui-ci pour l'accès aux données. Un exemple de réseau centralisé est donné par la figure 2.

Les serveurs web sont utilisés majoritairement selon ce mode : l'ensemble des clients connaissent l'adresse du serveur (par exemple <http://fr.wikipedia.org><sup>3</sup>) et effectuent les demandes de données à ce serveur qui se charge de renvoyer les informations demandés aux clients, prenant ainsi en charge la localisation et la distribution des données. Les systèmes centralisés présentent des avantages non négligables : la centralisation des ressources permet des recherches très expressives (par exemple par l'intermédiaire d'un langage de requête comme SQL ou XQuery) et dont le résultat est exhaustif (l'ensemble des données correspondant aux critères de la recherche sont renvoyées). De même, des solutions existent pour assurer la confidentialité et l'intégrité des données sur un système centralisé, car l'organisation en charge du nœud serveur peut définir la politique de son choix.

<sup>3</sup>Qui rencontre d'ailleurs actuellement des problèmes de passage à l'échelle, sa popularité allant grandissante ...

## Les limitations de la centralisation des ressources

- **Passage à l'échelle** La principale limitation vient de l'incapacité des serveurs à gérer un nombre élevé de clients et de ressources. En effet, la machine qui prend en charge l'exécution du serveur doit être suffisamment dotée en ressources de calcul, de stockage et de communication. Ceci entraîne une limitation des serveurs à quelques milliers de clients pour un coût important.
- **Sécurité et intégrité** Maintenir la sécurité d'un serveur est complexe. Les ressources centralisées sont facilement attaquables par des attaques de type déni de service (où l'attaquant va submerger de requêtes un serveur afin de l'empêcher de répondre à des requêtes régulières), de plus, il est difficile de maintenir l'intégrité des données, sauf en dupliquant dans des lieux différents géographiquement l'ensemble des données, ce qui présente également un coût élevé.
- **Utilisation du réseau physique** L'utilisation du réseau physique, qui permet de relier un nœud à un autre nœud quelconque, est perfectible. En effet, il existe un goulot d'étranglement dû à la capacité du serveur à répondre aux requêtes. La vitesse de réception des informations sur l'ensemble des nœuds dépend de la capacité et de la connexion au réseau physique de ce serveur, et pas de la capacité des liens entre les nœuds eux-même.
- **Contrôle et censure** Les systèmes centralisés permettent plus facilement un contrôle sur les nœuds qui peuvent accéder aux informations fournies par le serveur. Dans le cadre de régimes autoritaires pratiquant la censure, la propension des serveurs centralisés (pensons aux serveurs Web ou de diffusion de contenus) à être utilisés pour tracer ou filtrer les requêtes est un problème. Il est en effet relativement aisé de contrôler le point unique d'accès à l'information, qui est la plupart du temps clairement identifiable.

Il existe de nombreux cas de figure où ces limitations empêchent le déploiement d'applications réparties. Dans le cadre du partage de données, la capacité de passage à l'échelle vers des milliers ou des millions de participants est souhaitable, comme une utilisation plus efficace des ressources physiques ou la protection de l'anonymat des utilisateurs. Nous présentons dans la suite le paradigme pair-à-pair qui préside au fonctionnement des différents réseaux que nous abordons par la suite, et dont l'objectif est de répondre à ces besoins.

## 2.2 Modèle pair-à-pair

Le modèle de fonctionnement pair-à-pair est décentralisé. En ce sens, chaque nœud du réseau n'est pas fixé dans un comportement de client ou de serveur, mais chaque nœud peut agir à la fois comme client et comme serveur. De la même manière, les données présentes dans le système et que vont utiliser les nœuds ne sont pas regroupées sur un seul nœud mais peuvent être gérées par un nœud quelconque du réseau. Chaque nœud connaît un sous-ensemble de  $\mathcal{N}$  : ses voisins dans le réseau logique, auxquels ce nœud peut envoyer des messages.

Les propriétés attendues de ce modèle sont :

- **Auto-organisation** Le système ne nécessite pas d'intervention d'une entité externe pour s'organiser se-

lon les choix faits à sa conception. Par exemple, le choix des voisins d'un nœud est décidé de manière décentralisée.

- **Dynamisme** Les systèmes distribués pair-à-pair sont caractérisés par un taux de départ et de retrait de nœuds élevé. Il est important que l'insertion (placement d'un nouveau nœud dans le réseau) ou le retrait (qu'il soit volontaire, ou du à une panne) soient peu coûteux (que l'évolution de ce coût soit en  $\mathcal{O}(\log |\mathcal{N}|)$ ).
- **Équilibrage de charge et mutualisation des ressources** Contrairement au modèle client-serveur, l'espace mémoire et la capacité de traitement de requêtes sur les données augmente linéairement avec le nombre de nœuds, là où un serveur dédié au partage de données aurait été limité. Il est aussi possible d'équilibrer les traitements de requêtes en les répartissant sur l'ensemble des nœuds, qui partagent la même responsabilité.
- **Passage à l'échelle** Un corollaire des propriétés précédentes est qu'un système pair-à-pair présente souvent une propension au passage à l'échelle, contrairement aux systèmes centralisés. L'évolution des coûts de maintenance du réseau et de la charge imposée à chaque nœud participant, ainsi que d'autres mesures ne doit pas être linéaire en fonction du nombre de nœuds. On considère qu'un système passe à l'échelle quand par exemple ces mesures évoluent en  $\mathcal{O}(\log^x |\mathcal{N}|)^4$ .
- **Disponibilité et tolérance aux pannes** L'absence de centralisation et de point névralgique dans un système pair-à-pair le rend particulièrement intéressant pour le maintien de la disponibilité des objets (par exemple, en répliquant de manière active les données sur un certain nombre de nœuds du réseau) ou pour la tolérance aux défaillances : la défaillance d'un nœud quelconque dans un système pair-à-pair a moins d'impact que la défaillance du nœud serveur dans un système centralisé.
- **Contrôle, censure et anonymat** : De la même manière, l'absence d'entité de centralisation permet de proposer des systèmes visant à l'anonymat ou la résistance au contrôle par une entité externe tel le gouvernement autoritaire abordé plus haut. Nous ne développerons pas ce point, mais différents systèmes développés à cet effet existent : citons Free-Net [CMH<sup>+</sup>02] ou Mute.

## 2.3 Recherche dans un système pair-à-pair

L'objectif d'un système de partage de données, de ressources de calcul, ou de mémoire de masse, est de permettre à chaque participant de retrouver un ou des objets (données, type de ressource de calcul ou de mémoire de masse) correspondant à un ou des *critère(s)*. L'ensemble de ces *critères* forme une *requête*. Dans les systèmes centralisés, les requêtes des nœuds sont dirigées vers le serveur, responsable de l'ensemble des données, qui peut alors renvoyer un résultat exhaustif. Dans les systèmes pair-à-pair, la recherche ne peut pas se faire de manière

<sup>4</sup>l'augmentation de ces coûts est de plus en plus faible au fur et à mesure de l'ajout de ressources ou de nœuds dans le réseau.

aussi directe, car les données sont réparties sur l'ensemble de nœuds.

Un mécanisme de recherche de données dans un système pair-à-pair doit viser à présenter une partie des propriétés suivantes :

- **Expressivité des requêtes** L'expressivité décrit la richesse des critères possibles. On distingue différents types de critères : (i) ceux fondés sur le contenu des documents (par exemple les documents textes contenant un mot clé ou une chaîne de caractères conforme à une expression régulière) et (ii) ceux fondés sur les méta-données associées à chaque document (par exemple les documents dont les descriptions indiquent une taille de fichier comprise dans un intervalle). (iii) Une troisième classe, intermédiaire, peut être définie : les critères fondés sur les attributs. Les attributs sont des méta-données (par exemple des mots-clé) qui représentent le contenu, la sémantique du document. (iv) Enfin, l'expressivité minimale pour les critères est atteinte avec les critères sur clé. Chaque document possède une clé (par exemple un nom de fichier absolu) et il n'est possible de récupérer le document que si l'on connaît cette clé.

La recherche de données par critère fondé sur le contenu est difficile à mettre en œuvre de manière répartie, et utilise des méthodes qui dépassent le cadre de ce document. Nous nous intéressons donc principalement aux critères fondés sur les attributs et aux critères de type recherche exacte (par clé) dans des mécanismes de recherche répartis.

- **Exhaustivité des résultats** L'exhaustivité définit le taux entre la quantité de résultats obtenus par rapport au nombre de documents conformes à la requête et présents dans le réseau. Il est possible d'atteindre l'exhaustivité des résultats dans un système client-serveur car le serveur est responsable de l'ensemble de données. Toutefois, atteindre l'exhaustivité complète des résultats dans un système pair-à-pair peut revenir soit à interroger l'ensemble des nœuds, soit à réduire l'expressivité des requêtes ou leurs champs d'application.
- **Coût et efficacité** Un algorithme décentralisé de recherche dans un réseau pair-à-pair a un coût faible si le nombre de nœuds et de messages échangés pour obtenir la ou les réponses à une requête est faible. L'efficacité mesure le temps ou le nombre de messages nécessaires pour obtenir les résultats. Définir un algorithme de recherche peu coûteux ou/et efficace pour un système pair-à-pair implique des compromis sur l'exhaustivité et/ou l'expressivité proposée. De plus, comme pour les coûts de maintenance du réseau, il est nécessaire que l'évolution du coût de recherche ne soit pas linéaire en  $|\mathcal{N}|$ . Une propriété recherchée pour un tel algorithme est une évolution du coût en  $\mathcal{O}(\log^x |\mathcal{N}|)$ . Ainsi, une évolution polynomiale du nombre de nœuds dans le système se traduit par une évolution linéaire du coût de l'algorithme de recherche.
- **Autonomie** Un dernier critère que l'on peut prendre en compte est le respect de l'autonomie. L'autonomie est limitée si l'ensemble des nœuds que l'on va contac-

ter est défini strictement par le protocole. À l'inverse, l'autonomie est complète si un nœud peut décider d'envoyer sa requête seulement à un sous-ensemble de nœuds choisis dans le réseau, pour une raison quelconque. On peut par exemple penser à des nœuds qui souhaitent envoyer leur requête seulement aux nœuds correspondant à machines du même domaine (entreprise, université, ...). Nous ne centrons pas notre étude sur ce critère d'autonomie, mais il pourrait faire l'objet d'une étude approfondie.

### 3 RÉSEAUX STRUCTURÉS, RÉSEAUX NON STRUCTURÉS

Un réseau pair-à-pair peut être distingué selon l'organisation des nœuds (la topologie du réseau) utilisée pour la recherche et la localisation de données.

Le premier système de partage de données fondé sur le paradigme pair-à-pair, Napster [SGG03], repose sur une solution centralisée pour la recherche et la localisation de données, bien que l'échange de fichiers s'effectue par communications directes entre les nœuds.

Faire reposer le service de recherche et de localisation sur un serveur permet d'atteindre l'exhaustivité et une expressivité élevée, toutefois le système hérite des désavantages des systèmes centralisés évoqués plus haut. Le réseau Napster a été contraint à l'arrêt suite à une décision de justice, mais a permis à la communauté des systèmes pair-à-pair de saisir l'intérêt de mécanismes de recherche et de localisation de données décentralisés. En effet, la présence d'un point de centralisation pour la recherche rend le système entier aussi vulnérable que ne peut l'être ce nœud du réseau ; de plus cette solution montre clairement ses limites en termes de passage à l'échelle et d'utilisation des ressources.

Les réseaux pair-à-pair utilisant des algorithmes de recherche décentralisés peuvent être distingués selon deux types principaux, qui dépendent de la topologie du réseau logique (organisation des connexions entre nœuds dans le réseau logique). Les réseaux peuvent être structurés, totalement non structurés ou bien utiliser une approche hybride.

Dans un réseau non structuré, la topologie du réseau n'est pas fixée par une structure logique prédéfinie. Chaque nœud connaît un ensemble de voisins qui ne sont pas obligatoirement décidés par le protocole (même si leur sélection peut faire l'objet d'un choix en fonction de différents critères, comme nous le montrons à la section 7).

Les réseaux hybrides sont fondés sur une topologie hiérarchique : seule une partie des nœuds est utilisée pour le mécanisme de recherche.

Dans les sections suivantes, nous décrivons les principes et les propriétés de ces trois modèles, en donnons des exemples et en décrivons les capacités et les propriétés dans le contexte de la localisation de ressource.

## 4 RÉSEAUX LOGIQUES NON STRUCTURÉS

Dans un réseau logique non structuré, chaque nœud possède des liens vers un sous-ensemble de  $\mathcal{N}$ , ce sous-ensemble n'est pas déterminé par une structure logique précise (on peut prendre des liens vers des nœuds aléatoires). Nous présentons l'exemple du système pair-à-pair Gnutella, et montrons les mécanismes mis en œuvre pour la recherche de données.

### 4.1 Gnutella : recherche de données dans les réseaux non structurés

Gnutella fut le premier réseau pair-à-pair déployé totalement décentralisé, que ce soit pour la recherche comme pour le transfert des données. Succédant à Napster, dont la centralisation a causé la perte, Gnutella [Gnu] tira profit de cette expérience. La première version de Gnutella (v0,4) utilise pour la recherche un mécanisme d'inondation bornée [Rip01]. Un nœud  $n_1$  qui recherche une donnée envoie une requête à l'ensemble de ses voisins. Elle est assortie d'une borne sur le nombre de retransmission possibles :  $b$ . Dans le protocole original,  $b_0 = 7$ . Chaque nœud qui reçoit la requête effectue le traitement suivant : (i) il vérifie si une de ses données correspond à la requête. Si oui il envoie une notification à  $n_1$  ; sinon, si  $b > 0$ , il transmet la requête à l'ensemble de ses voisins, en décrémentant  $b$ . Si un nœud a déjà vu passer cette requête, elle est simplement ignorée.

La figure 3 montre un exemple de recherche dans un réseau Gnutella avec  $b_0 = 2$ , pour chaque étape de retransmission. Bien entendu, cette inondation n'est pas synchronisée, mais nous la présentons ainsi pour des raisons de simplicité.

- À l'étape  $t_0$ , le nœud  $n_1$  cherche une source pour la donnée  $a^5$ .
- à l'étape  $t_1$ ,  $n_1$  envoie la requête à l'ensemble de ses voisins, avec  $b = 2$ . La requête sera donc retransmise deux fois. Aucun des voisins immédiats n'a de donnée conforme à la requête.
- À l'étape  $t_2$ , les voisins de  $n_1$  (nœuds grisés) n'ayant pas répondu à la requête la transmettent à leur tour avec  $b = 1$ . Le nœud  $n_2$  possède la donnée  $a$ , il renvoie une notification à  $n_1$ .
- À l'étape  $t_3$ , une dernière propagation de la requête est effectuée avec  $b = 0$ . Le nœud  $n_4$  notifie  $n_1$  de sa possession de  $a$ .

Une fois que  $b$  atteint 0, la requête n'est plus retransmise. Toutefois, il existe des nœuds possédant des données conformes à la requête qui ne sont pas contactés : c'est le cas de  $n_3$ . Ainsi, la recherche par inondation bornée ne permet pas d'atteindre l'exhaustivité, elle n'assure d'ailleurs pas non plus qu'on trouve une donnée, même si celle-ci existe sur un des nœuds du réseau. Atteindre l'exhaustivité des résultats revient à interroger l'ensemble des nœuds, ce qui n'est bien entendu pas possible dans un système de cette taille. De plus,  $b_0$  étant fixé à 7, le nombre de nœuds contactés pour une seule requête est

<sup>5</sup>Le nœud  $n_1$  peut aussi envoyer une requête plus générale, par exemple récupérer des sources pour toutes les données dont l'identifiant commence par  $a$

élevé (dépendant du degré des nœuds, le degré moyen n'ayant pas de signification comme nous le montrons dans la partie 6.2), ce qui entraîne une charge réseau importante.

On remarque tout de même que l'expressivité est importante : il n'y a pas de limitation *a priori* sur les types de critères possibles pour une requête, les requêtes étant examinées par chaque nœud sur ses données propres. On peut ainsi utiliser des recherches par intervalles de valeurs, pour des sous-chaînes de caractères, ou tout types de critères complexes sur les méta-données<sup>6</sup>.

Le manque d'efficacité de ce protocole simple dû à l'inondation entraîne un problème de passage à l'échelle. Pour pallier ce problème (parmi d'autres), l'utilisation de réseaux hiérarchiques a été proposée (protocole Gnutella v0,6). Ceux-ci sont fondés sur l'utilisation de supernœuds, que nous présentons dans la partie suivante.

### 4.2 Réseaux hybrides et supernœuds

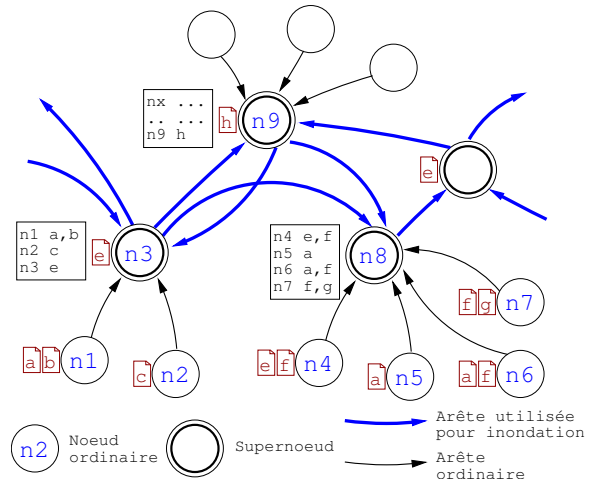


FIG. 4 – Réseau hybride : nœuds et supernœuds

L'un des réseaux pair-à-pair les plus utilisés pour le partage de fichiers est le réseau FastTrack, dont le client le plus connu est KaZaA [kaz]. Ce réseau est organisé de manière hiérarchique pour la recherche de données. La figure 4 présente un exemple d'un tel réseau. On distingue deux types de nœuds : les nœuds standards, qui ne prennent pas part au mécanisme de recherche, et les supernœuds qui prennent en charge la recherche de données dans le réseau.

Chaque nœud standard est connecté à un supernœud<sup>7</sup>. Le fait de devenir supernœud, pour un nœud standard, est décidé sur la base de ses caractéristiques (comme la mémoire disponible, la puissance du processeur, la présence de l'utilisateur devant la machine) ou sur son comportement (on sait par exemple que plus un nœud est resté longtemps connecté, plus la probabilité qu'il reste connecté une unité de temps supplémentaire est

<sup>6</sup>Il est aussi envisageable de proposer des critères sur le contenu même des documents, comme la recherche d'une chaîne de caractères dans les fichiers texte.

<sup>7</sup>Éventuellement plusieurs, ce qui ne change pas le fonctionnement de la recherche



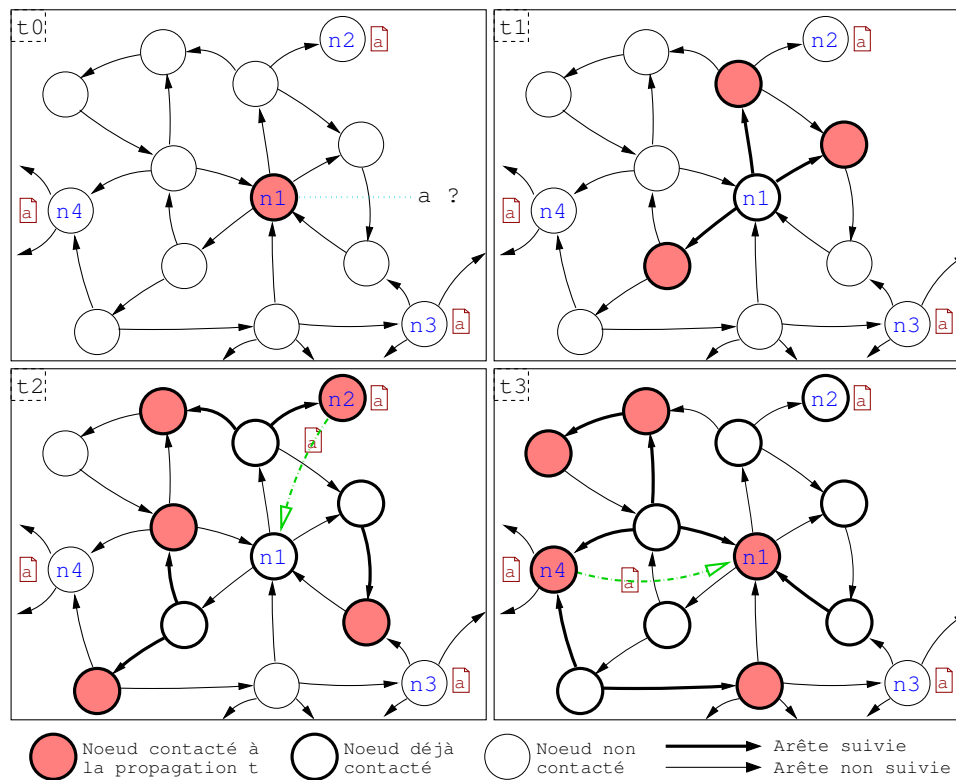


FIG. 3 – Recherche de données par inondation bornée

élevée [SPKG]). Les supernœuds sont élus parmi les nœuds selon ces critères.

Les supernœuds agissent comme des index : on voit par exemple dans la figure 4 que  $n_3$  conserve des informations sur les données que possèdent ses « feuilles » (les nœuds dont il s'occupe)  $n_1$  et  $n_2$ . La recherche est à l'initiative des supernœuds, soit pour eux-mêmes soit sur demande d'une de leurs feuilles. Elle est fondée sur le même protocole d'inondation que celui de Gnutella, décrit à la section précédente. L'inondation ne se fait qu'entre les supernœuds, qui renvoient les adresses des nœuds possédant les données (eux-même ou une de leurs feuilles).

**Expressivité et exhaustivité** Typiquement, le nombre de feuilles par supernœud est de l'ordre d'une soixantaine dans FastTrack et d'une vingtaine dans Gnutella v0.6. Le nombre de nœuds prenant part à la recherche est plus faible que dans Gnutella v0.4. La probabilité de trouver les données conformes à une requête par inondation bornée est donc plus importante, bien que l'obtention d'une donnée, même si elle existe dans le réseau, ne soit pas plus garantie que dans Gnutella v0.4.

Les supernœuds peuvent conserver dans leurs index les méta-données associées aux fichiers de leurs feuilles, ce qui permet des requêtes avec une bonne expressivité. Toutefois, il est plus complexe de mettre en œuvre la même expressivité que celle qui est possible dans Gnutella v0.4, car le volume de données à traiter peut devenir prohibitif sur chaque supernœud. Par exemple, il n'est plus possible d'envisager d'exprimer des requêtes sur le contenu

des données, et les méta-données ne doivent pas être trop complexes, pour des raisons d'espace mémoire utilisé sur le supernœud.

#### 4.3 Réseaux semi-centralisés, l'exemple de eDonkey

eDonkey [eDo] est un des réseaux pair-à-pair d'échange de fichiers les plus populaires, principalement *via* le client très répandu eMule. Il utilise une architecture semi-centralisée pour la recherche de données. Cette structure est fondée sur un ensemble de serveurs, dont les adresses sont connues de l'ensemble des nœuds. Ces serveurs ne prennent pas part au réseau d'échange : contrairement aux supernœuds FastTrack, il ne proposent pas de données. Toutefois, cette structure est similaire à celle des supernœuds dans le sens où un nombre restreint de nœuds est en charge de la recherche et de la localisation des données. Chaque nœud est connecté à un serveur, qui reçoit la liste des fichiers partagés assortie de leurs méta-données. Chaque serveur est en charge d'une moyenne de 50000 clients. Si un nœud souhaite effectuer une recherche sur tous les serveurs, il le fera lui-même en interrogeant les serveurs les uns après les autres (la liste des serveurs est connue).

Il est ainsi possible d'atteindre en théorie une quasi-exhaustivité<sup>8</sup>. En pratique, ceci rend le système de recherche du réseau eDonkey plus efficace pour trouver des fichiers rares : la probabilité de trouver un fichier peu partagé est nettement plus importante que dans le

<sup>8</sup>en réalité, il est difficile pour un nœud d'obtenir la longue liste de tous les serveurs, mais une liste de serveurs d'un millier d'entrée est courante.

réseau FastTrack. Le nombre de nœuds dont on examine le contenu des caches pour une requête est bien plus grand dans le cas d'un serveur que de quelques supernœuds. La centralisation de la recherche pose toutefois le même problème que celui qui s'est posé à Napster : la centralisation d'un service vital au réseau rend le réseau entier sensible aux attaques (techniques ou judiciaires par exemple). C'est pourquoi les créateurs de ce réseau ont proposé l'utilisation d'une table de hachage répartie pour limiter la charge de cet ensemble de serveurs [Ove, MM02]. Une table de hachage répartie est mise en œuvre sur un réseau structuré, dont nous présentons les principes dans la partie suivante.

## 5 RÉSEAUX LOGIQUES STRUCTURÉS

Les réseaux logiques structurés, à la différence des réseaux non structurés, imposent un lien entre les données dont est responsable un nœud et le placement de ce nœud dans la topologie. Cette topologie est inspiré de structures de données connues (arbres de recherche, listes chaînées avec des liens additionels : *SkipNets*, *Skip-Graph* [HJS<sup>+</sup>03], graphes de De Bruijn, ...). Les liens connectant chaque nœud à un ensemble de nœuds du réseau dépendent de cette structure.

Les réseaux structurés ont l'avantage de permettre une localisation rapide des données. En effet, si un nœud  $n_1$  cherche une donnée  $d$ , l'envoi d'une requête au nœud  $n_2$  qui est responsable  $d$  suit un protocole fixé par la topologie utilisée, le plus souvent en un nombre polylogarithmique en  $|\mathcal{N}|$  d'étapes. Par contre, cette efficacité est au prix d'une expressivité des requêtes plus faible, car fortement dépendante de la topologie choisie.

Dans la suite, nous expliquons les principes d'un réseau logique structuré mettant en œuvre une table de hachage répartie en prenant l'exemple d'une topologie en anneau, Chord [SMK<sup>+</sup>01].

### 5.1 Tables de hachage réparties

Une table de hachage répartie met en œuvre le principe d'une table de hachage : les données possèdent un identifiant (par exemple, dans un système de fichier, le chemin absolu `/home/toto/rapport.pdf`). Cet identifiant est envoyé à une *fonction de hachage* surjective, qui donne en retour une clé dans un espace de nommage de taille définie.

De la même manière, les nœuds possèdent une clé appartenant à l'espace de nommage. Cette clé peut être tirée aléatoirement avec une distribution uniforme, ou bien être le résultat du hachage de l'identifiant du nœud (par exemple son adresse IP). Le tableau 1 montre des exemples de hachage.

Cette fonction de hachage doit distribuer les clés de manière uniforme dans l'espace de nommage, ainsi pour un nombre d'éléments suffisamment grand (ce qui est le cas dans les systèmes considérés !) le nombre de clés par portion égale de l'espace de nommage sera à forte probabilité du même ordre de grandeur ; les nœuds sont répartis équitablement au sein de l'espace de nommage donc chaque nœud s'occupe à peu près du même nombre

d'objets, assurant ainsi un équilibrage de charge<sup>9</sup>. La taille de l'espace de nommage doit être grande en regard du nombre d'objets et de nœuds, afin de rendre faible la probabilité de collision de clés entre deux éléments différents.

Identifiant	Clé
mon_nœud1.free.fr:80	0x354F
217.12.3.11:80	0xFD01
/share/data/donnee_1	0x06D2
/share/data/donnee_2	0x3D10

TAB. 1 – Exemples de hachage des identifiants pour l'espace de nommage  $[0 \dots 2^{16}]$

Ainsi, les nœuds comme les objets sont représentés par des clés réparties uniformément dans le même espace de nommage. Les nœuds prennent en charge les données dont les clés sont proches de la leur. Cette proximité peut suivre diverses variantes dans sa définition, selon la topologie choisie. Il est important de noter que le nœud en charge d'une ou plusieurs donnée(s) peut ne garder que des liens vers le(s) nœud(s) possédant effectivement la donnée : il n'est pas toujours nécessaire ni possible de déplacer les données d'un nœud à l'autre.

Nous présentons dans le paragraphe suivant la table de hachage répartie Chord, qui est fondée sur une topologie en anneau.

### 5.2 La table de hachage répartie Chord

La figure 5 présente le principe de la table de hachage répartie Chord [SMK<sup>+</sup>01]. L'espace de nommage représenté est ici  $[0 \dots 30]$ , l'espace étant bouclé ; les auteurs proposent d'utiliser  $[0 \dots 2^m]$  avec  $m$  suffisamment grand (par exemple 128) pour éviter les collisions. Dans la suite, on parle d'anneau pour représenter l'espace des clés. Chaque nœud du système aura une identité appartenant à cet anneau. La figure 5 montre l'exemple de 10 nœuds et 11 données ainsi que leur placement sur l'anneau.

**Responsabilité pour les données** Chaque nœud est responsable d'une ou plusieurs données. Il s'agit pour lui de pouvoir renseigner un utilisateur cherchant une donnée en lui donnant sa localisation (l'adresse de la machine sur laquelle il pourra la trouver). Chaque donnée est associée à une clé (entier appartenant à l'espace de nommage). Le schéma 5 montre dans une même enveloppe les données et leur nœud responsable. Un nœud est responsable des données dont la clé précède directement la clé de ce nœud dans l'anneau. Ainsi,  $i$  et  $j$  appartiennent au nœud  $n_2$ .

### Liens et données maintenus sur chaque nœud

Chaque nœud a une place dans l'anneau selon sa clé. Il met à jour les informations concernant les données dont

<sup>9</sup>...minimal, car il faudrait aussi prendre en compte le nombre d'accès à cette donnée : s'occuper d'une donnée populaire est plus demandeur que de s'occuper d'une donnée peu recherchée !



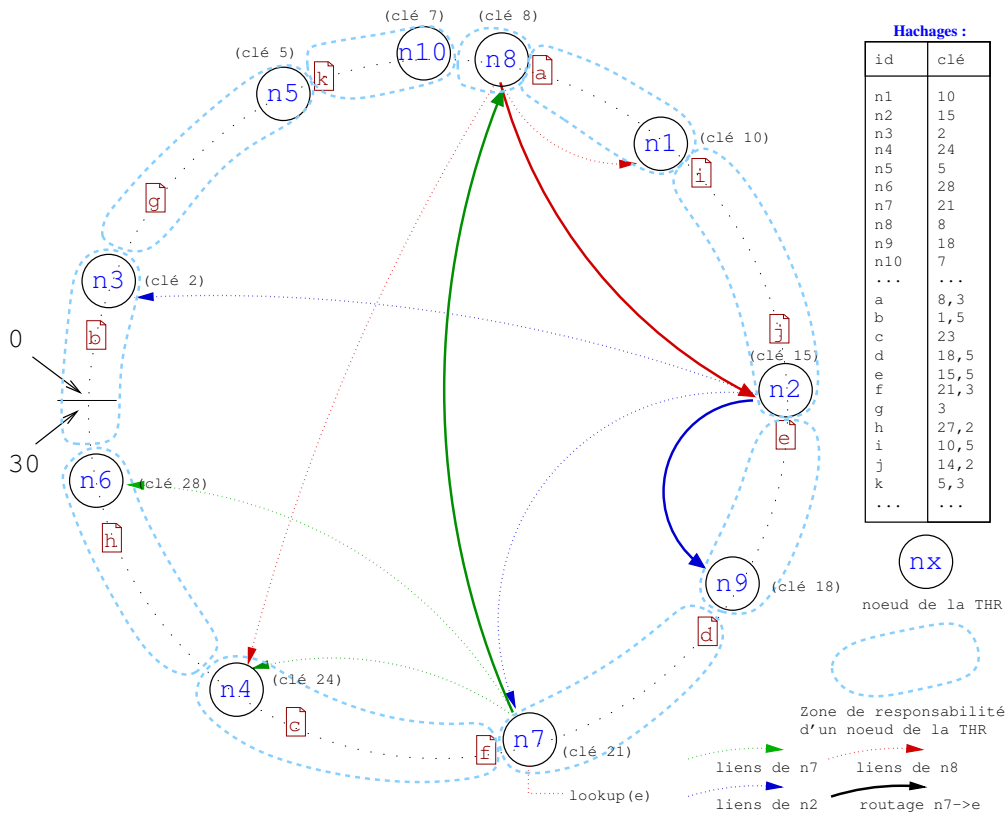


FIG. 5 – Chord (ne sont représentés que les liens pour  $n_2$ ,  $n_7$  et  $n_8$ ).

il a la responsabilité. La table de hachage donne une distribution homogène des clés sur l’anneau : chaque nœud est responsable en moyenne de  $\sim \frac{|D|}{|\mathcal{N}|}$  données.

De plus, chaque nœud  $n$  connaît son successeur immédiat dans l’anneau ainsi qu’un certain nombre de nœuds voisins appelés liens longs. Les liens de  $n_2$ ,  $n_7$  et  $n_8$  sont représentés sur le schéma 5. Le principe de ces liens est de permettre de « sauter » par dessus une fraction de plus en plus grande de l’espace de nommage.

Ainsi, le lien le plus long doit rejoindre un nœud situé aux antipodes du nœud  $n$ , les liens suivants couvrant un espace deux fois moins grand à chaque fois. La distribution des longueurs des liens est une distribution harmonique en terme de distance parcourue dans l’espace de nommage. Chaque nœud maintient  $\mathcal{O}(\log |\mathcal{N}|)$  liens.

**Routage entre un nœud et une donnée : localisation exacte** Lorsqu’un nœud souhaite accéder à une donnée, il doit en connaître l’identifiant. Dans l’exemple, le nœud  $n_7$  souhaite accéder à la donnée  $e$ .  $n_7$  applique la fonction de hachage à  $e$  et obtient  $h(e) = 15,5$ . Rejoindre le nœud responsable de  $e$  revient donc à router de manière gloutonne de  $n_7$  à  $n_9$ , responsable de  $e$ .

Le nœud  $n_7$  envoie sa requête à son voisin le nœud  $n_{suiv}$  (pointé par un de ses liens) dont la clé est inférieure à celle de  $e$  et qui minimise la distance  $\|n_{suiv}, e\|$ , qui à son tour retransmettra la requête. Grâce à la distribution harmonique des longueurs dans l’anneau, le nombre d’étapes pour router d’un nœud à une donnée est lui aussi en  $\mathcal{O}(\log |\mathcal{N}|)$ .

**Construction de la THR** Lorsqu’un nouveau nœud  $n_{nouveau}$  souhaite s’insérer dans l’anneau, il commence par hacher son adresse physique (ou tire une clé au hasard) pour obtenir une clé qui détermine sa position dans l’anneau. Il route ensuite un message vers le nœud  $n_{resp}$  responsable de cette clé ; ce nœud lui délègue alors les données qui sont dans sa zone de responsabilité, et devient le successeur de  $n_{nouveau}$ . Le prédécesseur de  $n_{resp}$  est notifié de son nouveau successeur  $n_{nouveau}$ . Les liens longs sont créés en routant vers les nœuds responsables de  $clé(n) + \frac{|\mathcal{E}|}{d}$ , où  $\mathcal{E}$  est l’espace de nommage et  $d$  varie de 2 à  $m - 1$ . L’opération d’insertion nécessite donc  $\mathcal{O}(\log^2 |\mathcal{N}|)$  opérations.

Si la THR est utilisée comme un index (les nœuds conservent des pointeurs vers les sources des données dont ils sont responsables),  $n_{nouveau}$  publie les données qu’il partage en routant leur description et son adresse physique vers les nœuds responsables pour les clés correspondantes.

La suppression d’un nœud est plus simple à mettre en œuvre : il suffit au nœud qui part de prévenir son prédécesseur et son successeur, qui mettent à jour leurs liens en conséquence.

### 5.3 Expressivité et exhaustivité des recherches dans les THR

La recherche dans une THR est exhaustive. En effet, le nœud responsable d’une donnée sait où se trouvent toutes les sources de celle-ci. Toutefois, l’expressivité est réduite dans le cas d’une utilisation directe du

mécanisme. Seule la recherche par un identifiant déjà connu est possible, et il n'est pas possible d'utiliser directement une recherche sur les méta-données des fichiers. Même si cet identifiant peut être obtenu à partir du contenu de la donnée ou de sa description, cela nécessite un moyen d'obtenir l'identifiant avant d'obtenir la donnée. De plus, les THR ne sont pas particulièrement adaptées à des recherches sur des intervalles de valeurs d'identifiant. En effet, le hachage ne préserve pas l'ordre des éléments, deux éléments d'identifiants consécutifs n'auront donc pas des clés successives dans  $\mathcal{E}$ . En ce sens, les tables de hachage réparties, si elles permettent d'obtenir l'exhaustivité et l'efficacité dans la localisation des données, ne permettent pas l'expressivité des réseaux non structurés<sup>10</sup>).

#### 5.4 Autres tables de hachage réparties

De nombreuses propositions de THR sont disponibles dans la littérature ; citons CAN [RFH<sup>+</sup>01], fondé sur un espace de nommage en tore euclidien à  $k$  dimensions, Pastry [RD01] et Tapestry [ZKJ01], fondés sur une topologie en hypercube prenant en compte la proximité dans le réseau physique lors de la création des liens, ou encore D2B [FG03], fondé sur une topologie en graphe de *De Bruijn*.

À notre connaissance, une seule THR est réellement déployée à grande échelle : il s'agit de Kademlia [MM02], qui est utilisée dans le protocole Overnet et mise en œuvre dans les versions récentes des systèmes pair-à-pair d'échange de fichiers eMule+ et eDonkey-hybrid [eDo]. Son utilisation permet de se passer des serveurs utilisés pour centraliser les localisations de données (voir section 4.3).

#### 5.5 Système de stockage répartis

Afin de pallier au problème de la recherche par localisation exacte des THR, de nouveaux systèmes que nous appellerons Systèmes de Stockage Répartis (SSR) apparaissent. Ils permettent, dans une certaine mesure, d'utiliser des requêtes par intervalles de valeurs ou des localisations de données dans un espace de nommage à plusieurs dimensions. Ne pas utiliser de fonction de hachage est alors nécessaire. Ceci complique les protocoles mis en œuvre, qui doivent alors assurer le bon comportement du routage et l'équilibrage de charge avec des mécanismes dédiés. Par exemple, le système Mercury [BAS04] utilise un anneau par dimension de l'espace de nommage. Il met en œuvre la répartition de charge et l'efficacité du routage à l'aide d'un protocole complexe de maintien des propriétés qui sont assurées par la fonction de hachage uniforme des THR.

*Des systèmes pair-à-pair aux réseaux d'interaction*  
Dans la partie suivante, nous présentons les réseaux d'interaction issus d'activité humaine. Les réseaux pair-à-pair

que nous avons décrit sont utilisés pour des applications qui sont, elles aussi, contrôlées par l'activité humaine : ses goûts, ses données populaires ou plus rares, ses regroupements de personnes autour d'intérêts communs ... Il nous a semblé intéressant de présenter de manière détaillée ce que sont les réseaux d'interaction, ce qui les caractérise et les éléments qui ressortent de leur étude dans de nombreuses disciplines. Cette étude est, bien sûr, menée dans l'objectif d'appliquer de nouvelles méthodes, de prendre en compte de nouvelles problématiques et de nouveaux modèles lors de la définition de mécanismes de recherche dans les réseaux pair-à-pair.

## 6 GRAPHES D'INTERACTION

Les graphes d'interaction sont une modélisation utilisée dans de nombreuses disciplines (même si dans le cadre de cette présentation, nous privilégierons les disciplines modélisant des comportements humains). À un instant de l'évolution du réseau, les acteurs sont les nœuds d'un graphe où une arête  $n_a n_b$  modélise une interaction entre  $n_a$  et  $n_b$ . Ce réseau est bien sûr dynamique, les nœuds peuvent être ajoutés ou supprimés durant son évolution. Les arêtes peuvent aussi être dynamiques, selon la définition de l'interaction considérée.

Une interaction, dans un tel réseau, est définie selon ce que l'on cherche à modéliser, par exemple :

- Si les acteurs sont des chercheurs, une interaction peut-être définie par : *a été coauteur pour au moins un article avec* (cette interaction est par nature bidirectionnelle et non dynamique)
- Si les acteurs sont des entreprises, une interaction pourrait être : *fait sous-traiter une partie de sa production par* (cette interaction est unidirectionnelle et dynamique)
- Si les acteurs sont des êtres humains, une grande quantité d'interactions peuvent être définies, comme par exemple : la connaissance (*connaît, a entendu parler de*) ; l'existence d'intérêts communs (*écoute le même style de musique, lit les mêmes auteurs ...*) ou encore l'existence d'une communication d'une durée minimale dans un passé proche : *pendant les  $t_p$  secondes précédentes, a parlé au moins  $t_{min}$  secondes avec ...*
- Enfin, si les acteurs sont des instances d'un système réparti pair-à-pair (les nœuds évoqués dans les sections précédentes) un indication de l'existence d'une interaction pourrait être : *a été notifié d'une réponse positive par* ou plus simplement : *peut envoyer un message à*<sup>11</sup>.

**Propriétés statistiques** Il est intéressant, pour les réseaux d'interaction de très grande taille, d'étudier de manière empirique ou analytique leurs propriétés, en particulier, dans des graphes où les acteurs sont des humains ou des objets dont le comportement dépend de décisions humaines. Il est raisonnable de penser que ces propriétés peuvent avoir un impact ou peuvent être utilisées pour améliorer l'algorithmique des réseaux pair-à-pair, dès lors que ceux-ci sont dédiés à un partage de

<sup>10</sup>Dans un réseau non structuré, il est possible de réaliser des requêtes complexes (par exemple sur les méta-données ou même sur le contenu des données) car la recherche va s'effectuer de manière locale sur un nœud, parmi les données qu'il possède (voir partie 4.1)

<sup>11</sup>Cette dernière interaction définit un graphe identique au réseau pair-à-pair considéré

données entre humains ou entre processus décidés par la volonté humaine.

La distribution des fréquences d'apparition des valeurs est une mesure intéressante pour de nombreuses propriétés. Nous distinguons deux distributions.

– **Distribution selon la loi de Poisson**

$$P(x) = \frac{e^{-\lambda} \lambda^x}{x!} \text{ pour } \lambda \geq 0 \text{ et } x = 1, 2, \dots$$

où  $x$  est le rang de l'événement dans la suite des événements ordonnée de manière décroissante selon la probabilité d'occurrence. Une distribution en loi de Poisson est caractéristique des distributions de probabilité centrées autour d'une valeur moyenne : on peut par exemple penser à la distribution des tailles des personnes. Si on divise les tailles possibles en paliers de 5 centimètres, on constate qu'il y a un grand nombre de personnes de taille proche à cinq centimètres près de la moyenne ; toutefois il n'est pas possible de trouver des personnes mesurant deux ou quatre fois la taille moyenne. La distribution est centrée sur la moyenne, il n'existe pas d'événement qui soit un ordre de grandeur plus grand que cette moyenne et dont la probabilité d'occurrence ne soit pas négligeable ;

– **Distribution selon la loi empirique de Zipf**

La probabilité d'occurrence d'un événement parmi une liste d'événements (ordonnés par ordre décroissant de probabilité d'occurrence) présente ces deux propriétés : (i) la probabilité des premiers événement dans la liste est élevée, il y a des événements très probables et (ii) la probabilité des événements décroît lentement, ce qui veut dire qu'il y a beaucoup d'événements possibles, même si leur probabilité d'occurrence est faible<sup>12</sup>. Un exemple caractéristique d'une distribution Zipf est la distribution des salaires dans la population active [Par96] : une fraction élevée de la population gagne un salaire faible ; il existe moins de personnes qui gagnent deux fois plus mais leur nombre est tout de même important, il existe encore moins de personnes qui gagnent 16, 32, 64 fois plus, mais il en existe. Si leur nombre baisse, il existe toujours une possibilité de trouver des personnes avec des salaires deux fois supérieurs.

Plus formellement, la fréquence d'occurrence d'un événement ( $P$ ) en fonction de son rang  $i$  (lorsque le rang est déterminé par la fréquence d'occurrence) est une loi de puissance

$$P_i \sim \frac{1}{i^\alpha}$$

où  $\alpha$ , constante très proche de 1, est la constante caractéristique de la distribution.

Exprimer la moyenne des valeurs des éléments pour un ensemble d'événements distribués selon une loi de Zipf n'a pas vraiment de sens. Les distribution Zipf sont aisément reconnaissables lorsqu'elles sont tracées

sur une courbe en log-log. Elles suivent alors le dessin d'une droite de coefficient directeur  $-\alpha$ .

D'autres exemples de distribution Zipf sont : les fréquences des mots dans un corpus quelconque de texte [Zip32] que le lecteur pourra expérimenter à [Gig] ou encore la taille des villes [Zip49]

## 6.1 Autres graphes d'interaction

À titre d'illustration de l'utilisation des réseaux d'interaction dans de nombreux domaines scientifiques, voici des exemples de ces réseaux dans des domaines et contextes variés.

- **Sciences sociales** : Les sociologues utilisent les grands réseaux d'interaction pour étudier les réseaux d'interaction sociale [Ber04]. Par exemple, il est intéressant de considérer le réseau des collaborations professionnelles, des amitiés ou des préférences sociales. À la frontière entre les sciences sociales et l'épidémiologie, l'étude du réseau des contacts sexuels permet de mieux saisir la dynamique de propagation des maladies sexuellement transmissibles. Par exemple, ce réseau montre une distribution des degrés (le nombre de partenaires) typiquement proche d'une distribution Zipf.
- **Sciences physiques** : Les physiciens s'intéressent aux modèles de propagation (d'épidémies ou de rumeurs par exemple [PSV01]), qui ne peuvent être étudiés sans prendre en compte le vecteur sur lequel se fait la propagation (réseau de relations sociales, réseau des transports de personnes ...).
- **Infrastructures** : Un grand réseau d'interaction connu et beaucoup étudié est le réseau des lignes aériennes, disponibles de manière fiable (ce qui n'est pas le cas des mesures empiriques de réseaux non cartographiés, comme nous le montrons plus loin pour les réseaux pair-à-pair de partage de données). Le réseau des transports est un réseau valué (chaque arête entre deux nœuds – deux villes – est pondérée par le flux de personnes y transitant) ce qui ajoute des problématiques particulières qui sortent du cadre de ce tutoriel.
- **Linguistique** : Les linguistes étudient des graphes d'interaction comme les graphes de co-occurrence des mots dans un texte ou les graphes de proximité sémantique issus de dictionnaires : chaque nœud est un des verbes de la langue étudiée, et une interaction est définie comme :  $n_1 \rightarrow n_2 \Leftrightarrow n_2 \text{ apparaît dans la définition de } n_1$ . Ces graphes sont de diamètre faible et on y retrouve des distributions Zipf, comme pour le nombre d'occurrence des mots dans un corpus de textes [Zip32].
- **Informatique** : Un graphe qui fait l'objet de nombreuses études est le graphe du Web. Il s'agit d'un graphe d'interaction qui définit comme nœud une page d'un site du Web, et l'interaction par :  $n_1 \rightarrow n_2 \Leftrightarrow n_1 \text{ contient un lien vers } n_2$ . L'étude de ce graphe, outre les intérêts commerciaux qu'elle présente (par exemple étudier la validité des algorithmes de classement utilisés par le moteur de recherche Google), permet de comprendre plus finement la structure et la dy-

<sup>12</sup>Une autre manière d'énoncer cette propriété est de dire que un tout petit sous-ensemble des possibilités arrive très souvent tandis qu'une très large fraction arrive rarement.

namique de l'Internet. Il présente aussi des intérêts sociologiques pour étudier la présence et les tailles de communautés virtuelles par exemple.

Ce graphe a de bonnes propriétés de partition (le coefficient d'agrégation de ce réseau est élevé : les nœuds ont un voisinage localement dense, où la probabilité d'avoir deux voisins eux-mêmes voisins est élevé ; cette caractéristique est typique des structures d'agrégat, c'est à dire de la présence de communautés) et une distribution des degrés suivant une loi de puissance [FFF99]. Il existe ainsi un petit nombre de pages auquel beaucoup d'autres sont liées (les moteurs de recherche par exemple) et une grande majorité de pages possédant peu de liens arrivant et partant d'elles.

## 6.2 Graphes d'interaction et graphes aléatoires

Afin de savoir si ces graphes présentent des propriétés particulières, il est intéressant de se demander si les propriétés observées de ces graphes d'interaction ne correspondent pas à celles d'un graphe aléatoire, où un nombre  $p \times |\mathcal{N}|$  d'arêtes  $n_1 \rightarrow n_2$  avec  $n_1$  et  $n_2$  tirés aléatoirement sont créés.

Un premier modèle simple de graphe aléatoire a été proposé par Erdős et Rényi [ER60]. Il engendre simplement un ensemble d'arêtes en tirant au hasard les nœuds à connecter pour chaque arête. Chaque arête a une probabilité  $p$  d'exister, il existe donc en moyenne  $p \times |\mathcal{N}| \times (|\mathcal{N}| - 1)$  arêtes dans le graphe aléatoire engendré. La création d'une arête ne dépend pas des autres arêtes précédemment créées.

Les graphes engendrés présentent les propriétés suivantes.

- **Distance moyenne faible** Pour toute paire de nœuds du graphe, il existe un chemin court entre ces deux nœuds.
- **Distribution des degrés** Les nœuds utilisés pour créer les arêtes sont tirés aléatoirement, la distribution du nombre de liens par nœuds suit donc une loi de Poisson centrée sur le degré moyen  $\frac{2 \times (|\mathcal{N}| - 1) \times p}{|\mathcal{N}|}$ .
- **Coefficient d'agrégation faible** Le graphe aléatoire d'Erdős-Rényi présente une transitivité faible ( $\sim p \times |\mathcal{N}|^2$ ). En effet, comme les liens sont créés aléatoirement, la probabilité que deux voisins d'un nœud  $n$  soient eux-mêmes voisins est égale à la probabilité que deux nœuds quelconques du graphe soient voisins. Il n'y a donc pas de phénomène de « communautés » ou structures d'agrégats qu'on puisse mettre en évidence dans un graphe aléatoire.

Or, l'ensemble des graphes d'interaction observés ne satisfont pas à l'ensemble de ces propriétés, ce qui montre que ces graphes ne correspondent pas à des graphes aléatoires. Watts et Strogatz [WS98] ainsi qu'Albert et Barabási [BA02] ont montré que les graphes observés satisfont les propriétés suivantes :

- **Distance moyenne faible** Comme pour le modèle aléatoire d'Erdős-Rényi, la distance moyenne entre les nœuds du réseau est faible.
- **Coefficient d'agrégation fort** L'ensemble des graphes d'interaction présentés ont un coefficient de transitivité forte. La probabilité pour un nœud que ses

voisins soient eux-mêmes voisins entre eux est non négligeable : le voisinage d'un nœud quelconque dans un tel graphe est donc « localement dense » : il y a des communautés de nœuds pour lesquels la connectivité entre eux est plus grande qu'avec le reste du graphe.

Il est aisé de se représenter cette propriété dans les réseaux sociaux, par exemple dans le réseau d'amitié. La probabilité que deux amis d'une même personne soient eux-mêmes amis est élevée.

- **Distribution des degrés, graphe sans échelle** Un graphe sans échelle est un graphe dans lequel la distribution des degrés des nœuds est très hétérogène [Bar02]. Une portion faible des nœuds est très fortement connectée tandis que la majorité est connectée à un petit nombre d'autres nœuds. La distribution des degrés est de type Zipf à décroissance rapide.

Ces nœuds fortement connectés jouent un rôle important dans l'étude de la topologie et de la dynamique des réseaux d'interaction. Par exemple, pour l'étude de la transmission de maladies contagieuses dans le réseau des contacts physiques ou sexuels, il faut prendre en compte leur rôle prépondérant dans la diffusion rapide de la maladie dans un graphe de diamètre moyen faible. La diffusion se comporterait de manière bien différente dans un réseau dans lequel la distribution des degrés des nœuds serait uniforme. Un autre exemple est l'importance des hyperonymes dans les réseaux d'interaction linguistique (un hyperonyme est un mot de sens plus général qu'un ensemble de mots : par exemple « *animal* » est hyperonyme de {*vache*, *cochon*, *poule*, ...}).

## 6.3 Réseaux petit-monde

Watts et Strogatz montrent que les propriétés précédentes se retrouvent dans des réseaux d'objets (issus de modélisations de la réalité) très variés, allant du réseau des neurones de l'ascaris *Cænorhabditis elegans* au réseau de distribution de l'énergie électrique aux États-Unis. Les réseaux possédant ces propriétés ont été nommé réseaux *small worlds* par Watts et Strogatz [WS98]. En anglais, l'expression « Small World ! » est l'équivalent de l'expression francophone : « Que le monde est petit ! ». Elle fut choisie pour désigner ces réseaux, afin de mettre en évidence le faible diamètre et la forte transitivité. Nous traduirons cette appellation par l'appellation « réseau petit-monde ».

Les réseaux petit-monde semblent présenter des propriétés intéressantes pour de nombreuses applications, et constituer un modèle efficace pour nombre des phénomènes étudiés. Toutefois, deux questions se posent lors de l'étude de ces réseaux petit-monde.

- **Pertinence** Les propriétés observées sont-elles mesurables empiriquement dans les réseaux concernés ?
- **Généralité** Un réseau quelconque peut-il être transformé en réseau possédant les propriétés des réseaux petit-monde ?

Nous présentons dans la suite un modèle de construction de réseaux petit-monde aléatoires, deux méthodes de « petit-mondisation » de réseau, ainsi qu'une me-

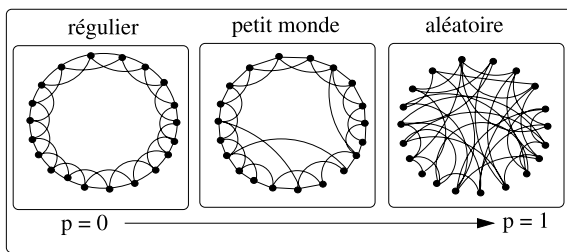


FIG. 6 – Construction de réseau petit-monde : processus de brassage de Watts et Strogatz

sure expérimentale : celle du diamètre du réseau des connaissances personnelles entre résidents des États Unis (expérience de Milgram).

**Modèle d'Albert et Barabási** Barabási et Albert ont proposé un modèle de génération de réseaux possédant les propriétés des réseaux petit-monde. Ce modèle est fondé sur une construction itérative du réseau (les nœuds sont insérés les uns après les autres) et sur le principe de l'attachement préférentiel : la probabilité qu'un nœud crée un lien avec un autre nœud est proportionnelle au degré de cet autre nœud. Chaque nœud possède le même nombre  $m$  de liens sortants. Ainsi, les nouveaux nœuds ont tendance à se lier à des nœuds déjà fortement liés dans le réseau ; ce phénomène s'observe par exemple dans la construction du graphe du web : les nouvelles pages ont tendance à posséder des liens vers des pages déjà fortement liées par ailleurs (moteurs de recherche, journaux, ...).

Ce modèle simple laisse supposer que les propriétés des réseaux petit-monde sont une loi de la nature ; ce modèle a donné lieu à de nombreuses études sur les réseaux, et le modèle a été raffiné pour prendre en compte des comportements des nœuds spécifiques à une discipline. Par exemple, les sociologues prennent en compte l'homophilie (attachement préférentiel à des nœuds de caractéristiques proches) et la distance d'intérêt lors de la construction de réseaux petit-monde aléatoires.

**Modèle de Watts et Strogatz** Watts et Strogatz ont montré que l'ajout d'un petit nombre de liens aléatoires peut transformer un réseau faiblement connecté en un réseau fortement connecté. Un réseau est faiblement connecté si son diamètre moyen est élevé. Par exemple, un réseau organisé en anneau est faiblement connecté : en moyenne, le plus court chemin d'un point à un autre est  $\frac{|N|}{2}$ . La méthode utilisée par Watts et Strogatz est de partir d'un réseau régulier (voir figure 6) où les nœuds sont organisés en anneau et connaissent leurs deux successeurs. Un pourcentage  $p$  de ces liens sont ensuite rebrassés, c'est à dire que la destination du lien au départ d'un nœud est changée : à sa place, un nœud destination aléatoire est choisi. Bien sûr, si  $p = 1$  on obtient un graphe aléatoire équivalent à ceux obtenus avec le modèle d'Erdős-Rényi.

Cette transformation d'un graphe régulier en un graphe petit-monde paramétré par  $p$  montre les implications des

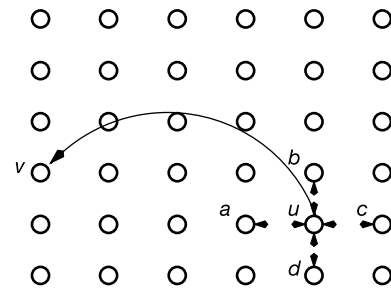


FIG. 7 – Construction de réseau petit-monde : modèle de Kleinberg, liens courts et lien long pour un nœud de la grille

propriétés petit-monde pour un réseau : (i) implications positives, par exemple en ajoutant judicieusement des routeurs dans le réseau des ordinateurs, on crée un grand réseau de communication (Internet !) qui présente un diamètre faible ; (ii) implications négatives par exemple pour le graphe de proximité sur lequel se propage un virus comme le SARS : chaque individu sain n'est distant que d'un faible chemin d'au moins un individu infecté, à cause du réseau de transport aérien.

**Modèle de Kleinberg** Le modèle de Kleinberg [Kle00a] est fondé sur une grille (espace à deux dimensions). Chaque croisement dans la grille est un nœud et chaque nœud connaît ses quatre voisins immédiats : haut, bas, gauche et droite. En plus de ces liens « courts », chaque nœud tire un lien long au hasard vers un autre nœud quelconque de la grille. La figure 7 montre un exemple de connexion pour un nœud de la grille. Ce modèle possède de bonnes propriétés algorithmiques [Kle00b] : comme les nœuds se situent sur une grille à deux dimensions euclidiennes, et qu'on connaît les coordonnées des nœuds destinations des liens courts et longs, il est possible d'envoyer un message entre deux nœuds en se fondant sur la seule connaissance locale des nœuds. Un nœud  $n_1$  décide en effet d'envoyer le message au nœud dont les coordonnées rapprochent le plus de la cible. Si  $c(n_x)$  est le couple  $(x_x, y_x)$  de coordonnées du nœud  $x$ , alors  $n_1$  choisit parmi les destinations de ses liens le nœud  $n_2$  qui satisfait :  $\|c(n_2), c(n_{dest})\|$  est le minimum pour l'ensemble des  $n_2$  possibles sur  $n_1$ . Le routage suit alors la propriété de faible diamètre du modèle petit-monde : il faut un nombre polylogarithmique d'étapes pour atteindre un nœud à partir de n'importe quel autre.

**Expérience de Milgram** Cette expérience a pour objectif de montrer le faible diamètre et l'aspect « sans échelle » des graphes petit-monde.

En 1967, Stanley Milgram, un psychologue de l'université de Yale conduit une expérience sur la distance moyenne entre deux personnes dans le graphe des connaissances proches entre les personnes disposant d'une adresse postale aux États Unis [Mil67]. Le principe de l'expérience<sup>13</sup> était de faire transiter une lettre entre

<sup>13</sup>...qui ne doit pas être confondue avec l'autre célèbre expérience de

des personnes de toute origines sociales situées à Wichita dans le Kansas vers la femme d'un de ses étudiants dans le Massachussets. La consigne donnée au participants était de faire transiter cette lettre seulement à une connaissance proche dont ils estimaient qu'elle était la plus apte à l'envoyer à la cible, que ce soit directement ou par l'intermédiaire d'une autre connaissance.

50 lettres furent distribuées et cet expérience mis en avant la propriété suivante : la médiane du nombre d'intermédiaires contactés dans la chaîne de transmission est environ égale à 6. Bien qu'il apparut plus tard que cette conclusion était fondée sur un faible nombre de lettres arrivées à destination, elle reste connue sous le nom des « 6 degrés de séparation ». Des expériences similaires, où une valeur (factice) donnée à la lettre permis d'augmenter considérablement le nombre de chaînes terminées, aboutirent toutefois à la même constatation.

Si cette expérience ne montre pas nécessairement que le réseau des connaissances entre personnes, pour toutes les personnes, est un petit-monde, il subsiste peu de doute qu'un tel réseau contenant *toutes* les personnes puisse être considéré comme un ensemble de réseaux petit-monde non disjoints.

Un autre résultat moins connu des expériences de Milgram est l'effet « Tunnel » : les personnes qui ont joué le rôle le plus important dans la transmission (par un meilleur rapprochement vers la cible et par une participation dans plusieurs chaînes) sont des personnes peu nombreuses et très fortement connectées par rapport aux autres. Ceci montre l'effet des nœuds de fort degré dans un réseau de type « petit-monde ». Si leur nombre est faible, leur degré est plusieurs ordres de grandeurs supérieur à celui des autres nœuds et ils jouent un rôle clé dans le réseau.

#### 6.4 Partition d'intérêt

**Transitivité du graphe d'interaction** Les graphes d'interaction sont caractérisés par un voisinage localement dense des nœuds. Il existe des communautés liées entre elles : si l'interaction définie dans le graphe est *partage des goûts avec* ou *est susceptible de s'intéresser aux informations du même type*, alors il est possible de définir des communautés fondées sur l'intérêt. Le réseau des proximités d'intérêt est transitif : la probabilité que deux personnes ayant un goût en commun avec une même tierce personne est élevée.

**Intérêt pour les systèmes pair-à-pair de partage de ressources** Comme il est décrit dans la partie 4.1, la recherche dans les réseaux pair-à-pair non structuré est fondée sur des processus d'inondation. Toutefois, le réseau logique sur lequel sont effectuées ces recherches ne dépend pas de l'interaction des nœuds dans l'application. Un nœud connaît un ensemble de nœuds du réseau, cet ensemble est choisi selon : (i) des critères de proximité dans le réseau physique, afin de connaître des nœuds pour lesquels la communication sur le réseau physique soit ef-

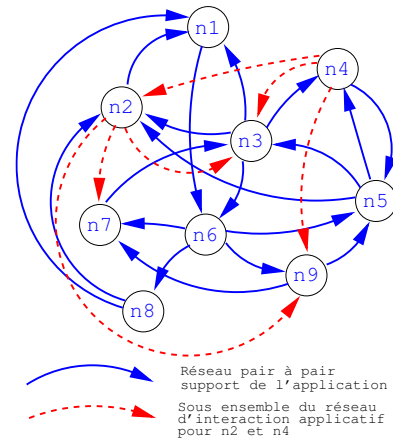


FIG. 8 – Illustration de la différence entre réseau pair-à-pair et réseau d'interaction applicatif (sous-ensemble)

ficace ; (ii) aléatoirement, en conservant  $m$  liens vers les  $m$  nœuds vus en premier après l'insertion.

Pourtant, le réseau pair-à-pair est le support d'une application : cette application implique des interactions entre les entités qui y participent. Par exemple, la figure 8 montre un réseau pair-à-pair non structuré de 10 nœuds. Les liaisons en lignes bleues pleines représentent les liens : par exemple l'adresse IP du nœud  $n_8$  n'est connue que de  $n_6$  ;  $n_8$  connaît les adresses IP de l'ensemble des nœuds  $\{n_1, n_2, n_6\}$ .

Ce réseau pair-à-pair est utilisé pour une application de ressources quelconques (données ou autres). Ainsi, on peut définir une interaction  $\mathcal{I}$  de plus haut niveau (au niveau *applicatif*) qui est :  $n_a \rightarrow n_b \Leftrightarrow n_a$  s'intéresse aux objets de même nature que ceux auxquels s'intéresse  $n_b$ . Les lignes discontinues rouges représentent une partie du graphe d'interaction selon  $\mathcal{I}$  : les liens pour les nœuds  $n_2$  et  $n_4$ .

On peut remarquer que les liens du réseau d'interaction, cette interaction étant définie par l'application, ne sont pas concordants avec ceux du réseaux pair-à-pair. Pourtant, l'interaction  $\mathcal{I}$  définit une proximité d'intérêt entre les pairs : la probabilité qu'ils aient à leur disposition des ressources qui intéressent l'autre est plus élevée que la probabilité qu'un nœud quelconque du réseau ait cette ressource à disposition.

Afin de passer à l'échelle et de faire une utilisation efficace du réseau physique, il est important pour un système pair-à-pair que les requêtes qui sont propagées ne transitent pas par un nombre important de nœuds. Or, pour atteindre les nœuds qui possèdent les ressources d'intérêt pour  $n_4$ , par exemple  $n_7$ , le nombre de nœuds à contacter en suivant les liens du réseau pair-à-pair ( $\{n_4 \rightarrow n_5 \rightarrow n_3 \rightarrow n_6 \rightarrow n_7\}$ ) est important : il faudrait ici que la borne pour le nombre de retransmission dans le processus d'inondation soit au moins de 4 pour espérer trouver la ressource, si elle n'est présente que à plus de 4 sauts à partir de  $n_4$ . Si l'on suit le même processus d'inondation dans le petit-monde qu'est le réseau d'interaction selon  $\mathcal{I}$ , le chemin est  $\{n_4 \rightarrow n_2 \rightarrow n_7\}$ . Ainsi, une borne de retransmission pour l'inondation de la requête sur le réseau d'interaction minimale est ici de 2 pour localiser la res-



source. En effet, la probabilité de trouver cette ressource dans les voisins immédiats et à quelques sauts dans le réseau pair-à-pair simple est assez faible.

Dans un système pair-à-pair où la connaissance d'adresses d'autres nœuds est aléatoire, seules les ressources répandues vont être facilement retrouvables par un nœud, car l'inondation sera obligatoirement bornée. Si la topologie du réseau pair-à-pair reflète une partie de la topologie du réseau d'interaction tel que défini par l'application, alors la recherche d'informations ou d'objets dans le réseau peut être rendue plus efficace.

**Proximité d'intérêt** Prendre en compte le graphe d'interaction revient dans le cas d'une application de partage de données à prendre en compte la *proximité applicative* : la proximité définit à quel point deux nœuds partagent des intérêts en commun. On parle aussi de *proximité d'intérêt* dans le cas précis d'objets définis par des méta-données, ou de *proximité sémantique*.

Prendre en compte la proximité d'intérêt permet aussi de retrouver le phénomène de communautés propre aux réseaux d'interaction. Si un ensemble de pairs disséminés dans le réseau partagent un intérêt pour les mêmes fichiers, alors la présence de liens directs entre ces pairs permet à chacun d'entre eux d'obtenir de manière plus efficace les données qui l'intéresse.

Bien entendu, les partitions d'intérêt que l'on peut obtenir ne sont pas disjointes : un nœud peut partager des intérêts différents avec plusieurs ensembles de nœuds. De même, la proximité d'intérêt peut aussi être liée avec d'autres proximités : par exemple, dans le cadre d'un système de partage de fichiers utilisé pour des films, la proximité d'intérêt est liée à la proximité géographique, pour des raisons simples de langue et de culture communes.

Dans la partie suivante, nous présentons les problématiques associées à l'utilisation des liens de proximité d'intérêt dans le cadre des réseaux pair-à-pair de partage de données.

## 7 UTILISER LES PROPRIÉTÉS DES GRAPHS D'INTERACTION POUR AMÉLIORER LA RECHERCHE DANS LES RÉSEAUX PAIR-À-PAIR

Les problématiques générales qui se posent au concepteur de systèmes pair-à-pair de partage de données sont les suivantes, s'il souhaite prendre en compte la proximité applicative dans la topologie de son réseau :

- **Intérêt ?** Existe-t-il un intérêt à prendre en compte la proximité d'intérêt dans cette application précise ? En effet, même si la proximité d'intérêt est mesurable, il est possible que la prise en compte de celle-ci ne soit pas aussi intéressante que la prise en compte de la proximité physique [LCP04] (latence d'envoi des messages sur le réseau IP par exemple). La prise en compte de la proximité applicative améliore-t-elle la qualité du mécanisme de recherche ?
- **Proximité ?** Le principal problème va être la définition d'une mesure de proximité : comme définir un indicateur qui estime que les intérêts de deux nœuds sont

proches ou non ? Cette mesure est-elle valide pour l'ensemble des couples de nœuds ou existe-t-il des couples dont la proximité d'intérêt importante ne soit pas capturée ? Inversement, assiste-t-on à l'émergence de faux positifs, la mesure de proximité estimant proches des couples qui ne partagent pas d'intérêts communs ?

- **Mesure ?** La mesure de cette proximité d'intérêt doit se faire localement, il n'est pas possible d'utiliser une connaissance globale du système pour prendre des décisions sur la proximité d'intérêt des nœuds. Quelle est la pertinence de la mesure locale par rapport à une hypothétique mesure globale de la proximité ?
- **Utilisation ?** Comment peut-on utiliser cette mesure de proximité pour adapter ou enrichir la topologie du réseau pair-à-pair ?
- **Dynamisme ?** Enfin, quel est le comportement dynamique de la prise en compte de la proximité d'intérêt ? Si les intérêts des nœuds pour les données changent régulièrement, la topologie est-elle modifiée en conséquence ?

Nous présentons quelques réponses existantes à ces questions dans les paragraphes qui suivent. Nous illustrons notre propos par un ensemble d'exemples récents de travaux proposés pour prendre en compte la proximité d'intérêt dans les réseaux pair-à-pair.

### 7.1 Existence de partitions sémantiques

Tout d'abord, il convient de mesurer la présence de proximités d'intérêt dans les réseaux de partage de données. Nous montrons dans la dernière section les méthodes existantes pour obtenir des informations sur des réseaux pair-à-pair déployés (collecte de *traces*).

Une première étude sur le réseau de partage de fichiers largement déployé<sup>14</sup> eDonkey [eDo] a été menée [FHKM04]. Elle consiste entre autres en l'analyse des intersections entre les contenus des caches de nœuds du réseau eDonkey.

Dans le système de partage de fichiers eDonkey, chaque client possède un nombre de fichiers qu'il met à disposition de la communauté, son cache. Chaque fichier est identifié par un hachage unique de son contenu (les fichiers ne peuvent pas être confondus). À chaque fichier sont associés des méta-informations qui décrivent son contenu, et qui servent à proposer des mécanismes de recherche expressifs, mis en œuvre au niveau des serveurs de recherche et de localisation. Ces méta-informations ne sont pas utilisés dans cette étude, seule la présence de communauté partageant les mêmes ensembles de fichiers a été étudiée.

La trace utilisée concerne 12000 clients partageant environ 923000 fichiers. L'étude montre la présence d'une forte proximité géographique pour les fichiers de type vidéo ; deux nœuds situés dans le même pays ont une plus grande probabilité de partager les mêmes fichiers. Pour les fichiers audio, l'étude montre la présence d'une forte partition d'intérêt : la probabilité que deux nœuds qui partagent une même donnée dans leurs caches partagent une autre

<sup>14</sup>plus d'un million de nœuds

donnée est élevée. Il existe donc un intérêt à prendre en compte la proximité d'intérêt dans de telles applications.

## 7.2 Définition du critère de proximité

La définition d'un critère de proximité, qui mesure à quel point deux nœuds sont proches, est une question complexe. Une mesure possible est celle de la taille de l'ensemble issu de l'intersection des contenus des caches des nœuds (leur ensemble de fichiers partagés). Toutefois, cette mesure simple n'est pas nécessairement significative : l'ensemble dans lequel sont intéressés les deux pairs peut être grand au regard de l'ensemble de fichiers partagés, et les pairs peuvent partager des fichiers très communs dans le réseau, qui vont faire monter leur critère de proximité d'intérêt sans que ce soit nécessaire.

Une autre approche possible est similaire à un cache de liens vers des nœuds considérés comme proches au niveau des intérêts [SMZ03, HKFM04]. Elle se fonde sur l'idée simple qu'un nœud qui répond positivement à une requête pour un fichier a une probabilité plus élevée de répondre positivement à une autre requête proche dans le temps qu'un nœud quelconque du réseau. Ainsi, chaque pair considère simplement que les nœuds proches au niveau des intérêts sont ceux qui ont répondu positivement à ses requêtes passées. Comme le degré sortant d'un nœud est borné pour des raisons d'espace mémoire disponible, les auteurs de [SMZ03] proposent de ne garder que les liens qui augmentent réellement les performances du mécanisme de recherche (qui répondent positivement aux requêtes du nœud considéré) en supprimant les liens peu utiles lors de l'ajout d'un nouveau lien.

Poussant plus loin ce concept, il a été proposé de créer un réseau construit à partir des requêtes effectuées par les pairs [FGL05]. La méthode de recherche est un parcours en profondeur guidé par le voisin de plus fort degré. Dès qu'un nœud découvre qu'un autre nœud a une donnée qui l'intéresse, il s'y connecte. Des simulations montrent que cette méthode permet de trouver les données en un nombre d'étape logarithmique, comme les THR, tout en permettant des requêtes beaucoup plus complexes (approximatives, intervalles, ...).

Il est aussi possible de définir une classification des documents (potentiellement hiérarchique) à la manière de celle utilisée dans les bibliothèques [CGM02]. Ainsi, il est relativement aisé de connaître les nœuds possédant des intérêts en commun en comparant les parties de la classification pour lesquelles ils partagent un intérêt. Malheureusement, la définition d'une telle classification statique nécessite que tous les pairs la connaissent, qu'ils ne trichent pas sur la présentation de leurs intérêts ; et il ne faut pas non plus que cette classification soit dynamique. Ces contraintes rendent cette solution peu intéressante pour les systèmes de partage de données à très large échelle car la nature de ces systèmes est d'être intrinsèquement dynamiques.

Enfin, une approche intéressante est l'utilisation des proximités d'intérêt issus d'une autre application répartie. Le système SPROUT [MGGM04] utilise les liens sociaux comme liens de proximité d'intérêt. L'idée de base est que des personnes utilisant un des nœuds du réseau sont

susceptibles d'avoir une proximité d'intérêt forte avec leurs contacts personnels sur le réseau Internet. Ces liens sont obtenus à partir de la liste des contacts d'un logiciel client des réseaux de messagerie instantanée, comme Jabber, MSN ou ICQ.

Toutefois, aucun de ces systèmes ne prend en compte la distribution de la popularité des données. La popularité peut être définie comme le nombre de répliques d'une donnée, c'est à dire le nombre de pairs en possédant une copie. En effet, il est sans doute pertinent de pondérer l'impact de la popularité (inversement de la rareté) des données considérées sur la mesure de proximité. La recherche de fichiers populaires bénéficie moins de l'ajout de liens sémantiques par rapport à la recherche de fichiers peu répandus. On peut donc imaginer ne garder des liens sémantiques qu'avec les nœuds pour lesquels il existe une proximité d'intérêt pour des fichiers peu populaires.

## 7.3 Mesure locale de la proximité d'intérêt

La mesure de la proximité est locale et non symétrique : il pourrait être intéressant de prendre en compte la mesure de proximité vue par les pairs destinations des liens sémantiques, afin de converger vers une mesure plus précise.

## 7.4 Prise en compte des proximités

Deux approches principales existent pour prendre en compte les proximités applicatives et la topologie du réseau d'interaction (réseau de proximité d'intérêt) : faire correspondre la topologie du réseau pair-à-pair avec celle du réseau de proximité d'intérêt, ou utiliser un réseau additionnel reflétant seulement la topologie mesurée du réseau de proximité d'intérêt.

**Remplacement des liens** On peut remplacer les liens du réseau pair-à-pair par les liens supposés (découverts) plus riches en proximité sémantique [Bus05]. Cette approche a pour but de faire converger la topologie du réseau pair-à-pair avec celle du réseau de proximité d'intérêt ; elle permet une gestion des liens plus simples mais elle peut être sujette à des partitions du réseau<sup>15</sup>, ou a une efficacité moindre si des recherches ne correspondant pas aux proximités sémantiques sont envoyées dans le réseau. En effet, le gain en efficacité pour les requêtes les plus courantes est associé à une perte d'éclectisme en ce qui concerne les intérêts des nœuds auquel l'émetteur est relié. Cet éclectisme est présent de manière naturelle dans le réseau pair-à-pair sans prise en compte des proximités applicatives.

**Réseau logique additionnel** Utiliser un réseau additionnel constitué uniquement de liens obtenus par mesure de la proximité sémantique, et émettre ou diffuser les requêtes sur ce réseau en priorité [LHSH04, VKMvS04, HKFM04].

<sup>15</sup>À cet effet, l'auteur de [Bus05] propose de conserver quelques liens vers des nœuds aléatoires, afin d'assurer les propriétés petit-monde du réseau.

Cette approche entraîne une plus grande complexité puisque chaque nœud doit maintenir deux listes de nœuds dans chacun des réseaux. Toutefois, le fait de pouvoir compter sur le réseau pair-à-pair non modifié peut être intéressant dans deux cas :

- Si un nœud envoie une requête sur des données qui ne correspondent pas à ses intérêts « habituels ». En effet, le risque peut être de créer des partitions selon les intérêts majoritaires si l'on fait correspondre la topologie du réseau d'intérêt et celle du réseau pair-à-pair. Le réseau pair-à-pair, de par le choix aléatoire (en regard des proximités d'intérêt), présente un éclectisme dans les intérêts des nœuds connus par chaque nœud, qui permet de pallier à ces effets négatifs. De plus, si la popularité des objets est prise en compte, alors il peut être plus efficace d'utiliser le réseau pair-à-pair classique, où des critères de choix des liens en fonction de la proximité dans le réseau ou la proximité géographique auront été utilisés. En effet, pour retrouver des objets très répandus, le gain d'utilisation des liens sémantiques est faible alors que le gain obtenu en terme d'utilisation du réseau physique peut être important grâce aux critères d'optimisation.
- Si les intérêts des nœuds changent régulièrement, alors il peut être difficile de réobtenir les nouveaux nœuds proches sémantiquement dans le réseau des proximités d'intérêt (intérêts précédents ...). À l'inverse, l'éclectisme dans les intérêts des nœuds connus dans le réseau pair-à-pair permet de retrouver plus aisément des nœuds possédant une proximité avec les nouveaux intérêts.

Il existe deux moyens principaux d'utiliser un tel réseau logique de proximité applicative supplémentaire :

- Si ce réseau logique est un réseau non structuré, on peut inonder ce réseau en premier avec la requête et si le nœud n'obtient pas suffisamment de résultats, utiliser une inondation sur le réseau pair-à-pair « standard ».
- Si ce réseau logique est un réseau structuré permettant des recherches exactes sur des clés, il est envisageable de l'utiliser pour publier les couples (identifiant ou mot clé, adresse du pair qui possède l'objet) pour les fichiers rares ; ce réseau logique peut être organisé en prenant en compte la proximité d'intérêt lors du choix des nœuds dans les tables de routage, dans la limite des contraintes fixées par le protocole.

## 8 OBSERVATION DE RÉSEAUX PAIR-À-PAIR

Pour comprendre et améliorer les mécanismes de recherche de données, il semble nécessaire d'observer les réseaux pair-à-pair déployés pour plusieurs raisons :

- Évaluer les propriétés de distribution des données ; quel est le profil de la distribution de popularité des données ?
- Évaluer les propriétés statistiques des requêtes envoyées (comme la popularité relative des différents attributs, la distribution des critères utilisés, selon les attributs ou encore la précision — à quel point les requêtes sont discriminantes pour les données du réseau —).

- Connaître le comportement dynamique des nœuds : quelle est la distribution des temps de connexion ? Les intérêts des nœuds changent-ils rapidement ou lentement ?
- Peut-on montrer l'existence d'une proximité d'intérêt ? (voir partie 7.1)

**Réseaux déployés** Les seuls réseaux pair-à-pair déployés à large échelle (où le nombre de nœuds dépasse un million) sont aujourd'hui les réseaux d'échange de fichiers. Selon les mesures obtenues sur [Sly] le nombre d'utilisateurs sur les quatre réseaux les plus répandus dépassait les huit millions. Le détail est montré dans le tableau 2. On remarque que l'application la plus déployée est eDonkey [eDo] qui se fonde sur une structuration semi-centralisée. Toutefois, les réseaux non structurés ou hybrides (Gnutella [Gnu] et FastTrack [kaz]) sont tout de même majoritaires en terme de nombres d'utilisateurs. Le réseau le plus récent, Overnet [Ove] est un réseau structuré utilisant un mécanisme de table de hachage répartie. Il dispose pour l'instant de peu d'utilisateurs mais ce nombre augmente régulièrement. Il est très probable qu'il supplante un jour les réseaux semi-centralisés comme eDonkey (avec lequel il est compatible), lorsque les serveurs de celui-ci auront été fermés pour une raison quelconque ...

Réseau (Client(s))	Utilisateurs
eDonkey (eMule)	3.778.383
FastTrack (KaZaa,Morpheus,Grokster)	2.659.955
Gnutella	1.584.984
Overnet (eDonkeyHybrid,eMule+)	573.440

TAB. 2 – Nombre d'utilisateurs moyen sur une journée de mai 2005 pour quatre réseaux d'échange de fichiers

**Pertinence et utilisation des mesures** L'observation de tels réseaux est malheureusement un problème très difficile : il n'est bien sûr pas possible d'obtenir des informations globales, et on est obligé d'être acteur du réseau observé. Les caractéristiques du réseau obtenues sont donc des extrapolations d'une observation locale d'un sous-ensemble du réseau.

Toutefois, même si les informations sont biaisées et ne représentent qu'une partie du réseau, il est intéressant d'en tirer : (i) des « traces » de requêtes et de contenu des caches des nœuds pour par exemple les utiliser comme entrée d'une simulation d'un nouveau mécanisme de recherche, et ainsi le confronter à des données proches du réel et (ii) d'en extraire des modèles probabilistes qui permettent de reproduire le comportement global du système autant de fois que nécessaire, pour essayer ou simuler une nouvelle méthode.

De la même façon, l'utilisation de traces peut être biaisée parce que les utilisateurs de logiciel de partage de fichiers vont faire preuve d'un comportement lui-même modelé par les possibilités du réseau. Utiliser de telles traces, obtenues d'un réseau où l'exhaustivité ne peut

pas être garantie (par exemple FastTrack) pour simuler un réseau visant à proposer une recherche d'expressivité limitée mais dont les résultats sont exhaustifs (typiquement une table de hachage répartie) n'est pas nécessairement représentatif de l'utilisation qui serait faite du réseau étudiée. Il est donc nécessaire de caractériser ce que représentent les traces obtenues, mais aussi ce qu'elles ne peuvent en aucun cas représenter. Quelques caractéristiques observées ne changent pas selon les traces étudiées, comme la distribution des temps de connexion, la présence d'agrégation ou la distribution Zipf des popularités des objets.

**Intérêt des mesures** Les communautés des systèmes répartis à large échelle et des réseaux pair-à-pair montrent un engouement important pour de telles traces, car la caractérisation d'une partie du comportement de telles applications déployées à une très large échelle échappe aux concepteurs d'applications et de protocoles. Utiliser des traces de systèmes réels pour la simulation de nouveaux protocoles est une solution complémentaire à l'utilisation de traces synthétiques (données engendrées selon une loi de distribution précise) pour la simulation ou l'émulation de réseaux de partage de données.

Certaines propriétés sont relativement difficiles à capturer avec un modèle : par exemple, l'agrégation des intérêts entre les nœuds, l'évolution de ces agrégations, la présence de partitions d'intérêt non disjointes et dont les tailles suivent elles-même une distribution Zipf (selon la distribution des popularités des objets éventuellement). L'utilisation d'une trace pallie ces difficultés, mais nécessite une attention particulière dans le choix de la méthode d'observation.

### 8.1 Méthodes d'observation et résultats

La méthode d'observation du réseau est fortement dépendante de sa structuration. Observer les requêtes et les publications (annonces par les nouveaux nœuds du contenu de leur cache) dans un système semi-centralisé comme eDonkey peut être efficace si l'on a accès aux informations transitant par le serveur. À l'inverse, observer un système non structuré ou hybride à base de supernœuds est plus complexe, puisqu'il faut être un acteur du réseau. Nous présentons ici quelques observations de réseaux pair-à-pair d'échange de fichiers présentées dans la littérature.

**Comportement des nœuds** Des études sur les aspects plus structurels du réseau ont été menées. Dans [SPKG, GDS<sup>+</sup>03], les auteurs présentent une évaluation du comportement des nœuds pour les deux réseaux les plus populaires à l'époque de l'étude, Gnutella (système non structuré, en 2002 sans utilisation de supernœuds) et Napster (recherche centralisée). L'étude a porté sur l'ensemble des communications entre les ordinateurs du réseau de l'université de Washington et le reste de l'internet. La population de nœuds est relativement homogène, tous sont des instances de l'application utilisés par des profils sociaux proches : étudiants et enseignants. De plus,

il n'est pas possible de voir les échanges qui se font au sein même de l'université, alors même qu'une proximité d'intérêt élevée est susceptible d'exister entre ces profils similaires. L'observation est faite au niveau des messages qui transitent par le réseau physique, ce qui ne permet pas de prendre en compte la sémantique des messages. Toutefois, leur étude a permis de montrer la grande hétérogénéité en terme de bande passante utilisée par chaque nœud ou en terme de quantité de données fournies au reste du réseau, et de montrer les aspects « sans échelle » dans le réseau décentralisé Gnutella : si les 4% de nœuds ayant les plus forts degrés sont supprimés, alors le réseau n'est plus connecté, et est divisé en plusieurs sous réseaux eux-même faiblement connectés.

Dans [AH00], les auteurs s'intéressent au phénomène de *free riding*, c'est-à-dire aux nœuds qui ne partagent aucune donnée et cherchent et reçoivent des données des autres nœuds. Ce travail a donné lieu à l'utilisation de moyens incitatifs pour les systèmes de partage de fichiers. Par exemple, dans le client eDonkey *eMule*, un système de score permet de coopérer avec les nœuds qui ont eux-même coopéré avec les autres.

### Partition et des intérêts et distribution des popularités

Cette approche est celle qui présente le plus d'intérêt dans le cadre de la prise en compte de propriétés des graphes d'interaction définis selon des proximités applicatives. En effet, elle prend en compte, au niveau du réseau pair-à-pair et des données, la sémantique des données partagées. Dans [HKM04]<sup>16</sup>, les auteurs montrent la présence de partitions d'intérêt entre l'ensemble des nœuds du réseau. Les courbes présentées sont fondées sur l'analyse d'une trace des contenus de caches de nœuds du réseau eDonkey. La méthodologie est la suivante : la topologie du réseau n'est pas simulée ; les nœuds peuvent mesurer le taux de résultat d'une recherche envoyée à des nœuds choisis aléatoirement ou à des nœuds choisis selon leur proximité sémantique. La mesure de proximité entre deux nœuds est l'intersection des ensembles de données présents dans leurs caches. Les requêtes utilisées sont obtenues à partir de la trace et sont des demandes de fichiers présents dans le réseau lors de la collecte de trace. Nous montrons quelques résultats qui représentent certaines des caractéristiques des réseaux d'interaction décrites à la section 6.4.

La figure 9 montre l'évolution de la proximité en fonction du critère défini. Elle montre que : (i) même si il existe peu de fichiers en commun, la probabilité de partager un autre fichier est importante (>35% au minimum) et (ii) la mesure de proximité permet de sélectionner des nœuds pour lesquels une amélioration des recherches de données est possible.

La figure 10 vise à montrer les apports de l'interrogation de nœuds proches selon la mesure de proximité d'intérêt à la place du même nombre de nœuds du réseau choisis aléatoirement. La liste de voisins proches sémantiquement est de taille bornée et gérée avec l'algo-

<sup>16</sup>utilisée pour les travaux [HKFM04] que nous présentons dans la partie 7.2

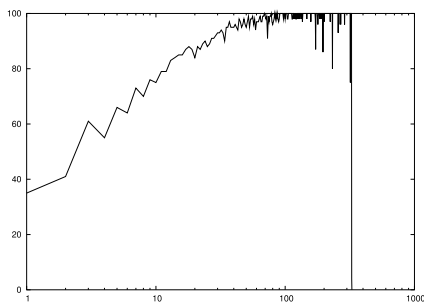


FIG. 9 – Mesure de l'évolution de la proximité : pour  $x$  fichiers en commun entre les caches de deux nœuds,  $y$  représente la probabilité d'avoir  $x + 1$  fichiers en commun.

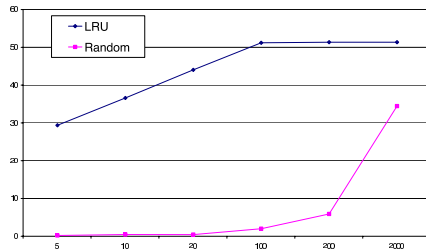


FIG. 10 – Interrogation de nœuds d'intérêts proches contre interrogation de nœuds choisis aléatoirement ( $x$  : nombre de nœuds contactés,  $y$  : taux de succès moyen des requêtes)

ritme LRU (lors de l'ajout d'un nouveau lien d'intérêt dans une liste pleine, le lien ayant le moins récemment répondu positivement à une requête est éliminé). On voit que le taux de succès est important dans le cas d'utilisation de voisins sémantiques ; toutefois, au delà d'une centaine de voisins sémantiques le gain n'est pas notable. Le taux de succès par demande à des nœuds aléatoires est très faible. Il faut alors interroger un nombre important de nœuds pour espérer obtenir un résultat.

La figure 11 montre la distribution des popularités selon le nombre de réplicats de chaque donnée dans l'ensemble du réseau ; la figure 12 détaille la distribution de cette popularité selon le type de fichier. On remarque que la distribution des popularités suit une loi de Zipf. On retrouve aussi des lois de Zipf pour les sous-ensemble considérés pour les distribution détaillées, typiques des caractéristiques des graphes d'interaction présentés.

**Proximités géographiques et musicales** Les courbes 13 et 14 présentent les distribution des langues et des genres des fichiers mp3 observés par les auteurs sur le réseau FastTrack à l'aide d'un supernœud modifié mis en œuvre au sein du logiciel MIDonkey [Fes02]. Les genres et langues sont décrits dans les entêtes *id3* des fichiers mp3 partagés par les nœuds. On constate là encore une distribution de type Zipf sur les langues utilisés (décrits sur deux lettres) ou sur les langues (selon la norme *id3*).

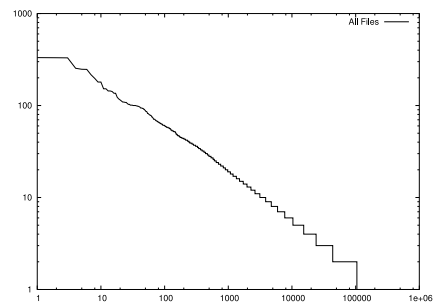


FIG. 11 – Popularités des fichiers en terme de nombre de réplicats dans les caches des nœuds du réseau. Courbe log-log ; en  $x$  rang du fichier dans la liste triée selon le nombre de réplicats ; en  $y$  le nombre de réplicats du fichier.

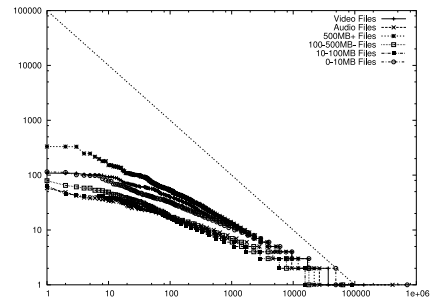


FIG. 12 – Popularités selon le type de fichier (même échelle que figure 11).

**Réseaux petit-monde** Dans [IRF04], les auteurs montrent la présence de comportement typiques des réseaux petit-monde dans les réseaux suivants : un réseau d'interaction où l'interaction est l'échange d'informations entre physiciens, les caches de données sur les serveurs Web et leurs clients et le réseau pair-à-pair d'échange de fichiers FastTrack. L'étude est fondée sur l'étude du graphe de partage de données, qui relie les nœuds (consommateurs) aux documents. Une interaction existe entre deux nœuds s'ils s'intéressent à la même donnée, ce qui est une définition de proximité d'intérêt. Les auteurs proposent de prendre en compte les caractéristiques petit-monde et la présence de communautés constituant elle-mêmes des sous-réseaux petit-monde lors de la conception d'algorithmes répartis de partage de données.

## 9 CONCLUSION

Dans ce tutoriel nous avons présenté deux objets d'études, les applications de partage de données décentralisées et les graphes d'interaction, qui présentent des caractéristiques communes intéressantes. Nous avons mis en avant certaines de ces caractéristiques, et présenté les approches actuellement envisagées pour tirer parti de ces propriétés dans le cadre des mécanismes répartis de recherche de données. Enfin, nous avons exposé la problématique de l'observation de tels réseaux et celle de l'exploitation des données partielles obtenues.

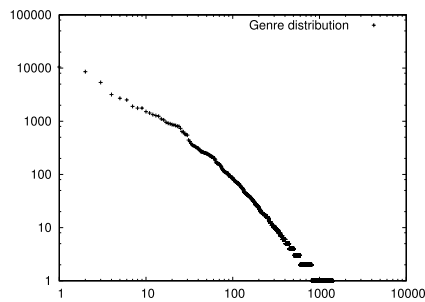


FIG. 13 – Distribution des genres des fichiers MP3 :  $x$  est le rang du genre dans la liste triée par la popularité,  $y$  le nombre de fichiers déclarant ce genre comme description.

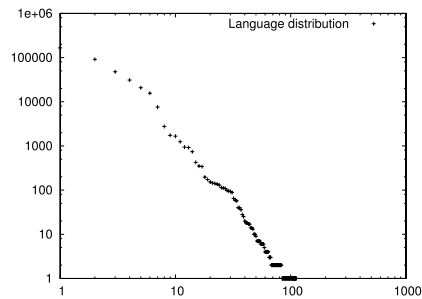


FIG. 14 – Distribution des langues des fichiers MP3 :  $x$  est le rang du langage dans la liste triée par la popularité,  $y$  le nombre de fichiers déclarant ce langage dans leur description.

## 10 PERSPECTIVES

Comme on l'a vu, des propriétés comme les communautés, la connexité et les distributions des objets commencent à être mises à profit pour créer des réseaux plus efficaces. Cependant, un important travail d'étude reste à effectuer. Certains réseaux d'interaction présentent des comportements différents, sans que l'on puisse modéliser finement les caractéristiques qui les différencient. Il est probable que dans le cadre du partage de données pour des applications de calcul réparties à grande échelle, il soit nécessaire de caractériser les différences de comportement de partage par rapport aux applications de partage de fichiers « grand public » présentées.

Un autre axe d'étude à défricher est la prise en compte du dynamisme dans de tels réseaux. Quel est le comportement des mécanismes de prise en compte de la proximité applicative face à des intérêts des nœuds changeant régulièrement ?

Découvrir les caractéristiques de tels réseaux permettrait de proposer un modèle, qui servirait alors de base à des simulations ou même des preuves des protocoles proposés, sans avoir à se servir de traces rares et difficiles à obtenir. À cet effet, il peut être intéressant pour les chercheurs en informatique d'« interagir » avec les autres scientifiques qui utilisent des réseaux d'interaction pour modéliser leurs objets d'étude. Cette pluridisciplinarité est, à notre sens, une des voies possibles pour mieux comprendre et modéliser le comportement de tels réseaux.

Du point de vue de la recherche en informatique, l'ob-

jectif est de proposer des protocoles qui soient efficaces, en prenant en compte les propriétés énoncées, mais qui se fondent sur une vision locale sur chaque nœud et prennent en compte les aspects dynamiques. Ces objectifs ne peuvent être découplés des autres buts qui animent la communauté pair-à-pair, comme l'augmentation de l'expressivité et de l'exhaustivité, ou encore les aspects d'anonymat ou de sécurité que nous n'avons pas abordé ici.

**Remerciements** Nous tenons particulièrement à remercier les auteurs de [FHKM04] et [HKFM04] pour avoir aimablement fourni les figures 9, 10, 11 et 12, issues de leurs travaux. Ce tutorial est issu de réflexions des auteurs dans le cadre de l'école sur les GRI organisée par le Groupement de Recherche Interdisciplinaire sur les Grands Réseaux d'Interaction (GRIGRI). Nous tenons à remercier ses organisateurs et les participants pour la richesse des échanges qui s'y sont déroulés.

## BIBLIOGRAPHIE

- [AH00] E. Adar and B.A. Huberman. Free riding on gnutella. *First Monday* 5, October 2000.
- [BA02] A.-L. Barabási and R. Albert. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 2002.
- [Bar02] A.-L. Barabási. *LINKED : The New Science of Networks*. Perseus Books Group, 2002.
- [BAS04] A. R. Bharambe, M. Agrawal, and S. Seshan. Mercury : supporting scalable multi-attribute range queries. In *SIGCOMM*, pages 353–366, 2004.
- [Ber04] R.L. Beriger. *Handbook of Data Analysis*, chapter The Analysis of Social Networks, pages 505–526. Sage Publications, 2004.
- [Bus05] Y. Busnel. Prise en compte de la proximité sémantique dans la restructuration de réseau logique pair à pair. Master's thesis, IRISA / ENS Cachan - Antenne de Bretagne, Rennes, France, juin 2005.
- [CGM02] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Stanford University, 2002.
- [CMH<sup>+</sup>02] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley. Protecting free expression online with Freenet. *IEEE Internet Computing*, 2002.
- [eDo] edonkey. <http://www.edonkey2000.com/>.
- [ER60] P. Erdos and A. Rényi. On the evolution of random graphs. *Publ. of the Math. Inst. of the Hungarian Academy of Sciences*, 5 :17–61, 1960.
- [Fes02] F. Le Fessant. Mldonkey, a multi-network file-sharing client. <http://www.mldonkey.org/>, 2002.



- [FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proc. of SIGCOMM'99*, pages 251–262, 1999.
- [FG03] P. Fraigniaud and P. Gauron. The content-addressable network d2b. Technical Report 1349, LRI, Univ. Paris-Sud, France, jan 2003.
- [FGL05] P. Fraigniaud, P. Gauron, and M. Latapy. Combining the use of clustering and scale-free nature of user exchanges into a simple and efficient p2p system. In *Proc. of Euro-Par'05*, 2005.
- [FHKM04] F. Le Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massoulié. Clustering in peer-to-peer file sharing workloads. In *Proc. of IPTPS*, 2004.
- [GDS<sup>+</sup>03] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, modeling and analysis of a peer-to-peer file-sharing workload. In *Proc. of SOSP*, 2003.
- [Gig] E. Giguet. Loi de zipf : applet java. <http://users.info.unicaen.fr/giguet/java/zipf.html>.
- [Gnu] Gnutella.
- [HJS<sup>+</sup>03] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, , and A. Wolman. Skipnet : A scalable overlay network with practical locality properties. In *Proc. of USITS'03*, 2003.
- [HKFM04] S. Handurukande, A.-M. Kermarrec, F. Le Fessant, and L. Massoulié. Exploiting semantic clustering in the edonkey p2p network. In *Proc. of SIGOPS*, 2004.
- [IRF04] A. Iamnitchi, M. Ripeanu, and I. Foster. Small-world file-sharing communities. In *Proc. of INFOCOM*, Hong Kong, march 2004.
- [kaz] Kazaa. <http://www.kazaa.com/>.
- [Kle00a] J. Kleinberg. Navigation in a small world. *Nature*, 406, 2000.
- [Kle00b] J. Kleinberg. The small-world phenomenon : An algorithmic perspective. In *Proc. 32nd ACM Symposium on Theory of Computing*, 2000.
- [LCP04] Eng Keong Lua, Jon Crowcroft, and Marcelo Pias. Highways : Proximity clustering for scalable peer-to-peer network. In *Fourth International Conference on Peer-to-Peer Computing*, pages 266–267, 2004.
- [LHSH04] B. T. Loo, R. Huebsch, I. Stoica, and J. Hellerstein. The case for a hybrid p2p search infrastructure. In *Proc. of IPTPS*, 2004.
- [MGGM04] S. Marti, P. Ganesan, and H. Garcia-Molina. Dht routing using social links. In *Proc. of IPTPS*, 2004.
- [Mil67] S. Milgram. The small world problem. *Psychology Today*, 2 :60–67, 1967.
- [MM02] P. Maymounkov and D. Mazieres. Kademlia : A peer-to-peer information system based on the XOR metric. In *Proc. of IPTPS*, March 2002.
- [Ove] Overnet. <http://www.overnet.com/>.
- [Par96] V. Pareto. *Cours d'économie politique*. Droz Genève, 1896.
- [PSV01] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86 :3200, 2001.
- [RD01] A. Rowstron and P. Druschel. Pastry : Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of Middleware*, 2001.
- [RFH<sup>+</sup>01] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of ACM SIGCOMM Technical Conference*, August 2001.
- [Rip01] M. Ripeanu. Peer-to-peer architecture case study : Gnutella network. In *Proc. of PDP'02*, August 2001.
- [SGG03] S. Saroiu, K. P. Gummadi, and S. D. Gribble. Measuring and analyzing the characteristics of napster and gnutella hosts. *Multimedia Syst.*, 9(2) :170–184, 2003.
- [Sly] Slyck, file sharing news and informations. <http://www.slyck.com/>.
- [SMK<sup>+</sup>01] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. In *Proc. of SIGCOMM'01*, 2001.
- [SMZ03] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proc. of INFOCOM*, 2003.
- [SPKG] S. Saroiu and S.D. Gribble P. Krishna Gummadi. A measurement study of peer-to-peer file sharing systems. In *Proc. of MMCN'02*.
- [VKMvS04] S. Voulgaris, A.-M. Kermarrec, L. Massoulié, and M. van Steen. Exploiting semantic proximity in peer-to-peer content searching. In *Proc. of FTDCS*, 2004.
- [WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of small world networks. *Nature*, 393 :440–442, 1998.
- [Zip32] G. K. Zipf. *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press, 1932.
- [Zip49] G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Harvard University Press, 1949.
- [ZKJ01] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry : An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, U. C. Berkeley, 2001.